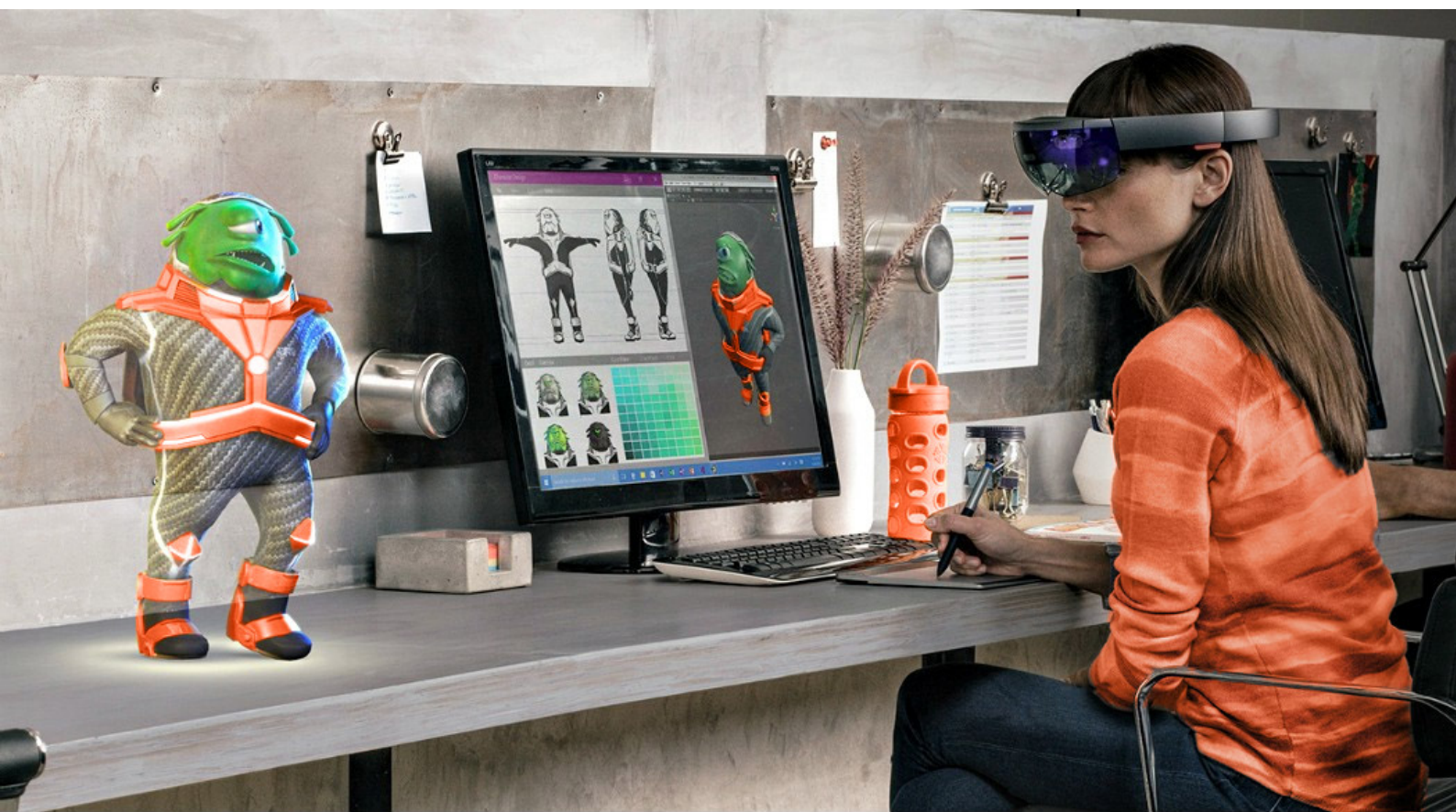


Superhuman Sports

Bachelor Thesis

T. Buckers
B. Gong



Superhuman Sports

**VRabl, a multiplayer augmented reality sports
game for the Microsoft HoloLens**

Bachelor Thesis

by

**Tim Buckers
Boning Gong**

in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science

at the Delft University of Technology,
to be defended publicly on Tuesday July 4, 2017 at 11:00 AM.

Project duration:	April 24, 2017 – July 4, 2017
Supervisor:	Prof. Dr. E. Eisemann
Client:	Dr. S.G. Lukosch
Thesis committee:	MSc O. Visser Dr. H. Wang

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This thesis is the result of our final Bachelor project and concludes our Bachelor of Science in Computer Science at the Delft University of Technology. The goal of this project was to apply the Software Engineering skills that are acquired during the Bachelor. It also let students demonstrate their ability to take on a real world assignment from a client.

In the past nine weeks we have been creating a Superhuman Sports game using the Microsoft HoloLens. For nine weeks we worked closely with our client, the Delft University of Technology, in order to ensure that the game meets the requirements, and to see what the possibilities are for such a Superhuman Sport. Furthermore, this project is supported by the Sports Engineering Institute of the TU Delft as well as the Participatory Systems Initiative of the TU Delft. This thesis aims to inform the reader about our initial research, design process, implementation and future work of this project.

We would like to thank everyone who helped us in making this project a success. In particular, we would like to thank our client, Stephan Lukosch, for taking time to weekly meet us and coming up with ideas and giving feedback on every prototype. We would also thank Otto Visser, one of the coordinators of the Bachelor project, for his fast replies and immediately tackling problems when they arose. Furthermore we would like to thank Elmar Eisemann for coaching us, the other coordinators for setting up the Bachelor projects and the students who participated in our user study and provided us valuable feedback.

*Tim Buckers
Boning Gong
Delft, June 2017*

Executive Summary

Console gaming is getting more and more popular and this can prevent people from being physically active. Physical activity is healthy for many reasons, including cognitive functioning and general development. By combining the popularity of video games and the necessity of physical activity we try to answer the following question: “How can we create attractive games, involving more physical activities, in a world where there exists a plethora of console games that are fun to play without any physical effort?” To answer this, we divide it into subquestions. We will start with making a game attractive. Literature shows that there are multiple factors that can make a game fun. Challenge, fantasy and curiosity are the three main components that a game needs to be fun.

One of the goals of the project is to enhance human capabilities. Augmented reality is a suitable method to achieve this. We have two Microsoft HoloLenses at our disposal. With these lenses we can add virtual objects to the real world. This offers the opportunity to design a real life game. The augmented reality aspect of the game helps players to get immersed into the game. When designing the game, we need to take into account that it must stimulate physical activity. Seamlessly integrating the movement into the game is important to make the game appealing to a broad audience. The physical aspect becomes less mandatory and more natural when players are immersed into the game.

Our game, VRable, is an adapted version of dodgeball, enhanced with augmented reality. It can be played in any open area inside or outside. The HoloLens first scans a place and then places a playing field. Both players have virtual balls that they can shoot at the bunnies at the other side of the field. The goal of the game is to hit enough bunnies and be the first to reach a certain score. Every bunny is worth a certain amount of points. Smaller ones will give more points than bigger ones. Players can also defend their bunnies by catching the ball. There are extra components such as power-ups that give players the opportunity to use different strategies. Lastly extra spatial sounds and special effects are added to make the game more appealing and engaging.

The user study shows that the game is fun to play and contains physical activity. Some of the more competitive players that got really immersed in the game were even weary after playing the game for less than ten minutes. There was also some criticism, mostly on the HoloLens. Users told us that it was heavy and had a limited view angle. One player commented that the game itself could be improved by paying more attention to visual details.

Overall we can say that VRabl is a good step forward for creating an attractive game that involves physical activity and can compete with other video games that does not require any physical effort. In the future when better augmented hardware is available and more affordable, games like VRabl can play a part in making physical activity more attractive. Especially for the younger generation, where many children and teenagers lack physical activities due to video games. In the past years, eSports became increasingly more popular as alternative to other spectator sports, particularly among younger viewers. We think that, with the expected technological developments and enough stimulation, console games with physical activity and Superhuman Sports can grow and start their own world wide competitions.

Glossary

Sports games - Games that involve physical activity and skill in which a team or individual competes against one another or others for fun.

Video games - Games that require some sort of user input and gives visual feedback, often played for fun and sometimes competitive.

Superhuman Sports - Superhuman Sports are defined as activities that (1) rely on technology for human augmentation to enhance a human ability, (2) involve physical fitness and skills and (3) are played for fun, competition or health reasons.

Virtual reality (VR) - An environment that is totally virtual which means that it is generated by one or multiple computers without any physical elements.

Augmented reality (AR) - A view of a physical, real-world environment with projected (augmented) elements by computer-generated sensor inputs such as sound, video or other data.

International Festival of Technology (IFoT) - Yearly festival in Delft. The main purpose is to promote and encourage technological developments and make people more aware about them.

Software Improvement Group (SIG) - Group of software experts that give organizations and companies advice about maintaining and improving software quality.

Automated Static Analysis Tools (ASATs) - Tools that will automatically analyze your code without actually running it. They give warnings in order to ensure code quality, style and maintainability.

MoSCoW method - A prioritization technique, among other things used in software development. The capitalized letters stand for the four prioritization techniques, Must have, Should have, Could have, and Won't have.

Spatial sound- When wearing the HoloLens, the audio engine provides the aural component of the mixed-reality experience by simulating 3D sound using direction, distance, and environmental simulations. This will give users a better perception of what is happening around them, without having to see it.

Hologram- When wearing th HoloLens, the 3D-objects that are visible are called holograms.

Unity - Unity is a cross-platform game engine. It is primarily used to develop video games and simulations for computers, consoles and mobile devices.

Unified Modeling Language (UML) diagram - Diagrams that are used with object oriented software engineering. It shows the structure and coherence of the software architecture.

TU Delft DreamTeams - Multidisciplinary team of mostly students from Delft working on certain machines/concepts. This can be everything ranging from submarines to cars racing on sunfuel.

CHI PLAY - The ACM SIGCHI annual conference on Computer-Human Interaction (HCI) in Play (CHI PLAY). The conference aims to foster discussion of current research in games and HCI as foundations for future digital play.

Contents

Preface	i
Executive Summary	ii
Glossary	iii
1 Introduction	1
2 Problem definition	2
3 Game concept	3
3.1 Requirements	3
3.2 Easy to understand	3
3.3 Physical activity	3
3.4 Engagement and fun	4
4 Implementation	5
4.1 Unity game engine	5
4.2 Programming C# and game components	6
4.3 Microsoft Sharing Service	7
4.4 Photon Unity Networking framework	8
5 Code quality assurance	9
5.1 Software Improvement Group	9
5.2 Automated Static Analysis Tools	9
5.3 Testing our system	10
5.4 Version control	10
6 Final product	11
6.1 Timeline of the game	11
6.2 Game field	11
6.3 Players and their controls	12
6.4 Shooting balls and their behaviour	12
6.5 Targets	13
6.6 How to win the game	13
6.7 Picking up power-ups and activating them	14
6.8 Different playing strategies	14
6.9 The value of sounds	15
7 Game evaluation	16
7.1 International Festival of Technology	16
7.2 User study and results	16
7.2.1 Approach.	16
7.2.2 Core module	17
7.2.3 Post-game module	17
7.2.4 Specific questions and their relevance.	18
8 Conclusion	19
9 Future work	20
9.1 Required improvements	20
9.2 From Superhuman Sports DreamTeam to the Tokyo Olympics	20
9.3 CHI PLAY student game design competition	21

10 Discussion	22
10.1 Ethical aspects that need to be considered	22
10.2 Project team	22
10.3 Automated integration testing	23
10.4 Augmented technology	23
10.5 Release of the game	23
A Research report	24
A.1 Introduction	24
A.2 Problem definition & analysis	24
A.2.1 Specific requirements	24
A.2.2 Fun factor	25
A.2.3 Augmented reality	27
A.2.4 Exertion Games Lab	29
A.3 Possibilities	29
A.3.1 Ideas	29
A.3.2 Enhancement with augmented reality	30
A.3.3 Downsides	31
A.3.4 Comparison	31
A.4 Plan of action	32
A.4.1 Organization	32
A.4.2 Code quality assurance	33
A.4.3 Testing	34
B User study questionnaire	35
B.1 Game Experience Questionnaire – Core module	35
B.2 Game Experience Questionnaire – Post-game module	36
C User study results	38
C.1 Table with ratings	38
D Software Improvement Group evaluation	41
E Requirements using the MoSCoW approach	42
F User manual	43
F.1 Starting the game	43
F.2 Game Menu	43
F.3 Standing ready	43
F.4 In-game	43
F.5 After game	44
G Process evaluation	45
G.1 Planning	45
G.2 External communication	45
G.3 Internal communication	45
Bibliography	47

Introduction

Games are an integral part of everybody's life. Some would say it is irreplaceable in their current life. There are many types of games ranging from video games to outdoor sports games. Games in general are activities with specific rules in which a person can participate. Video games are electronic games that involve interaction with a user interface to generate visual feedback on an electronic device. Technological advancements allow for more engaging and realistic video games. New technological concepts like augmented reality bring extra dimensions to video games. All sorts of video games are becoming increasingly more popular, especially among teenagers. These games are catching up the classical form of sports games, which are defined as games that include some sort of physical activity. We see a trend of people participating less in sports games and instead more in playing computer games. This leads to an increasing amount of people with health problems [21].

Several researchers in Japan have founded the Superhuman Sports Society and created the concept of Superhuman Sports. These are activities that rely on technology to enhance a human ability, involve physical fitness and skills and are played for fun, competition or health reasons. There are plans to introduce this concept during the Summer Olympic Games in 2020 in Japan. Towards this goal, several Superhuman Sports design competitions are planned with the first competition in July 2018 in Delft. This project will try to address the issue of increasing health problems due to a lack of physical activity by creating a Superhuman Sport game.

The game will encourage people to be more physically active. By combining competitive and video gaming aspects the game will make sure that people are having fun while being active at the same time. People will get immersed in the game and forget that they are physically active while playing it.

This thesis consists of multiple components, starting with our problem definition in Chapter 2. This chapter will define the problem and state our research question. In Chapter 3 we will describe our concept and explain how it addresses the problems defined in the previous chapter. In Chapter 4 we go more in-depth on the technical parts of how the game is built. It will elaborate on the used game engine, our development environment and setting up servers for our game. Chapter 5 will explain how good code quality in this project is maintained. Then Chapter 6 will give a full description of what the final product has become. Starting with a general timeline and thereafter explaining the important aspects separately. To validate that the product does address the problems, we did some evaluation in Chapter 7, for which a user study was conducted. In Chapter 8 we give the conclusion of this project. After that, we will give recommendations and explain what should happen in order to make Superhuman Sports successful in the future in Chapter 9. Subsequently Chapter 10 discusses about ethical aspects, the team and the technology.

2

Problem definition

Console gaming is getting more and more popular. Partly to the increasing convergence of new technologies and their possibilities [3]. Console games that are played competitive and for money are called eSports. In the past years, eSports became increasingly more popular as alternative to other spectator sports, particularly among younger viewers [7]. League of Legends, a popular eSport, had 43 million viewers during the final match of the 2016 World Championship [2]. This shows how hard the popularity of eSports and therefore console gaming is increasing.

There are clear downsides to this shift of popularity. One study showed that children who play moderate amounts of electronic games weigh more than their counterparts that played less [21]. There is also a significant positive relationship between physical activities and cognitive functioning [18]. Physical activities are a great part of their general development, especially for young children and teenagers. Most video games don't require physical activity, thus many of the gamers become less physically active.

It is clear that physical activity is good for us. The research question that we try to answer in this project is: "How can we create attractive games, involving more physical activities, in a world where there exists a plethora of console games that are fun to play without any physical effort?" To answer this question, we will try to create a game involving physical movement. One where the physical efforts are seamlessly integrated in the game. The game must also be fun and engaging to really fulfill the main goal.

3

Game concept

The goal of this project is to create a concept game that answers the research question. This chapter will elaborate on what the requirements are, how we came up with the idea and how our concept tries to answer the research question.

3.1. Requirements

When creating this game there are several requirements and restrictions we need to take into consideration. The game must involve physical activities and skills and should be played for fun, competition or health reasons. The focus of the project is on enhancing human capabilities, augmenting senses and merging realities. The sport needs to be a competition, involving some kind of information technologies to augment the players. This can be wearable or on the sports field. It needs to be able to be played on an indoor sports field. One round should last less than 30 minutes, including the time to explain the sport. During this time all aspects of the Superhuman Sport need to be presented. It can be a team or one versus one, game however it needs to be possible to compete on some level. The rules should be simple enough that it can be explained in 5-10 min to people with diverse backgrounds. For this project there are 2 HoloLenses available for development. This is a good option for the project, because they offer a suitable form of augmentation.

3.2. Easy to understand

To ensure the game is easy to understand we first looked at other already existing games and sports. A new game could be hard to understand and explain in a short period of time. The researched games during the research phase were mostly focusing on games and sports where augmented reality could really add value. We filtered the possible concepts based on the amount of requirements that were met. After finding out which fun inducing aspects could work in augmented reality we came up with our first concept. Considering the limitations of the hardware and tools, we created a concept based on dodgeball. Dodgeball is a familiar sport for most people, but is not played very much. We are able to tweak this concept in order to make it more exciting. It is also easy to understand for people who are familiar with it. For the group that is not familiar with dodgeball the concept is still easy to grasp. In our concept we replaced the throwing of balls, what normally happens in dodgeball, with shooting balls, like in shooters. This makes it more feasible to implement in the limited time we have.

3.3. Physical activity

This concept automatically contains physical activity, due to the similarity with dodgeball. In our concept the player has to catch balls instead of dodging. This small change in the dodgeball concept still forces the player to move around in order to play the game properly.

The player has to defend his/her targets by catching the balls, this requires movement. Furthermore, the shooting and aiming also requires head and arm movements.

3.4. Engagement and fun

When the game is fun to play and people immerse in the game the physical activity becomes less mandatory and more naturally. Therefore the fun factor is important and probably the most difficult requirement to fulfill. To make this project succeed people must engage in the game. Based on literature (see Appendix A) we are aware of the different aspects in a game that make it fun to play. While designing the game we will take the following aspects into account: challenge, fantasy and curiosity. Those aspects should add to a more engaging game. In Appendix A we go more in depth on these aspects and how to achieve this.

4

Implementation

This chapter we will go more in-depth in the technical aspects of VRabl. We will explain which IDEs we used, the structure of our game and the setup to let the HoloLenses share data.

4.1. Unity game engine

VRabl is programmed using the Unity game engine. This is a cross-platform game engine, which is often used to develop video games for computers, mobile devices and other consoles. It also has the option to develop augmented reality games for the Microsoft HoloLens. Programming a game in Unity, instead of building a game from the ground up, saved us lots of time.

Figure 4.1 shows how the game engine looks like. Unity works with scenes. Each scene contains its own objects and scripts. The screenshot below shows how our game scene looks like. It looks empty, this is because most objects will be spawned after the room has been scanned.

Often Unity scenes consists of many objects called prefabs. Those are types of assets/reusable game objects stored in project view. At startup our scenes already contain the prefabs that are needed at that moment. At run-time it is also possible to initiate prefabs in the scene. Most objects are spawned at runtime in the game scene. Objects can be shared over a server. This way the other player sees the same objects at the same locations.

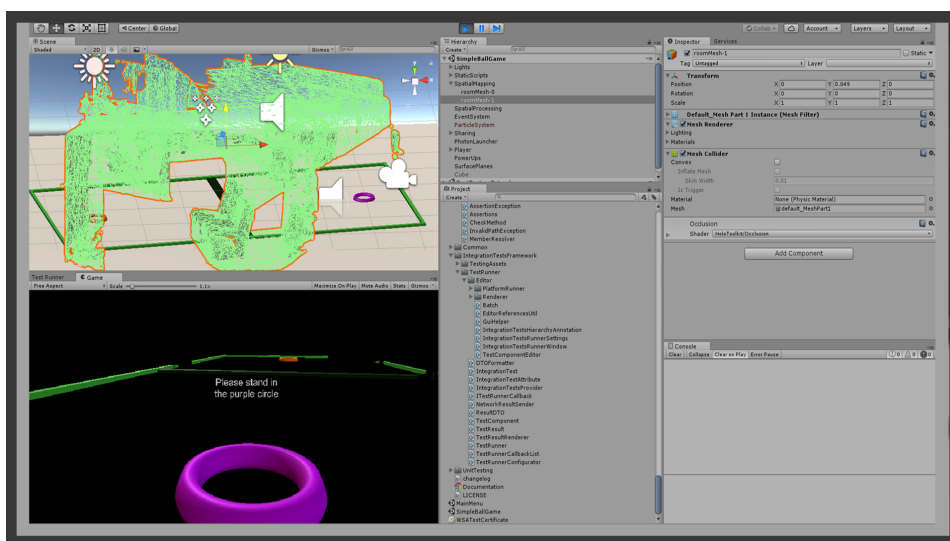


Figure 4.1: View of the Unity game engine and our game whilst in development.

The behaviours of the objects are determined in MonoBehaviour C# files. Here we can write code that will be executed every frame or is triggered by external calls. Those scripts will define how objects behave. When a carrot is spawn, a bunny should walk towards it. That is done by the RabbitBehaviour script for example.

4.2. Programming C# and game components

All scripts are written in C#, using the Microsoft Visual Studio IDE. Some classes are just plain C#, like the classes described in Figure 4.2. Other classes control certain Unity objects, like all classes within the Behaviour map (see Figure 4.3.A). The map is created with our custom C# code. The placement of the map is done by first analyzing the current room. Microsoft offers code that takes care of the scanning of the room. Eventually all found surfaces are analyzed to find the proper location on the floor for the map (see Figure 4.3.B). During the game the heads-up display (HUD) provides the player information. The heads-up display consists of three components: one for general information, another for keeping track of the amount of balls the player has left and the last one to display the current score (see Figure 4.3.C). On start up the game has a menu with three simple buttons. By gazing at a button and performing the tap-gesture a button is clicked (see Figure 4.3.D).

Then there are scripts that control the connection between the game and the Microsoft Sharing Service and Photon network, we will elaborate more on those later in this chapter.

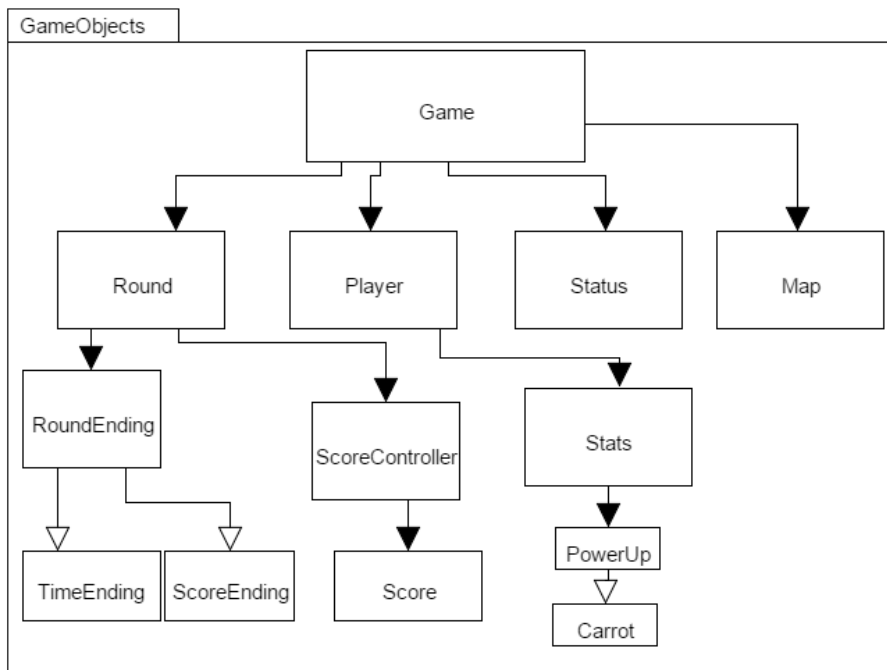


Figure 4.2: A clear overview of all the classes that make up the game.

We have an AppStateManager that keeps track of the current state of the game. We distinguish the following states:

- Starting
- GettingIp
- Connected
- Scanning
- ReceivingMap

- CreatingMap
- StandingReady
- ReadyToStart
- InGame
- EndGameht

Those states indicate the state of the game. Every state has their own functions that are called accordingly. GettingIp for example will call the class that listens to the voice input of the user. Scanning will enable the scanning mode of the HoloLens, so it is able to scan the room and gather the data. Based on this data, the first player will get in the CreatingMap state and places the map on the right place. The second player will go from the Scanning state directly to the ReceivingMap state. It will get the placed map of player 1 by communicating with the Microsoft Sharing Service.

4.3. Microsoft Sharing Service

An important aspect of this two-player game is sharing data. Most objects in the scene need to be visible for both players. When the map is placed, both players must see the map at the same real world position. To achieve this we use the Microsoft Sharing Service. This service is built for the HoloLens and has build in features such as sharing holograms (see Figure 4.4). Unfortunately there was the problem that sharing moving objects gave a huge delay. That is why we chose to use the Photon Unity Networking framework to synchronize the moving balls.

Another problem of the Microsoft Sharing Service is that the server IP changes when you switch network connection. Both players must know the IP before they can join the same server. Previously there was an auto-join option, but due to pull request #69 on the

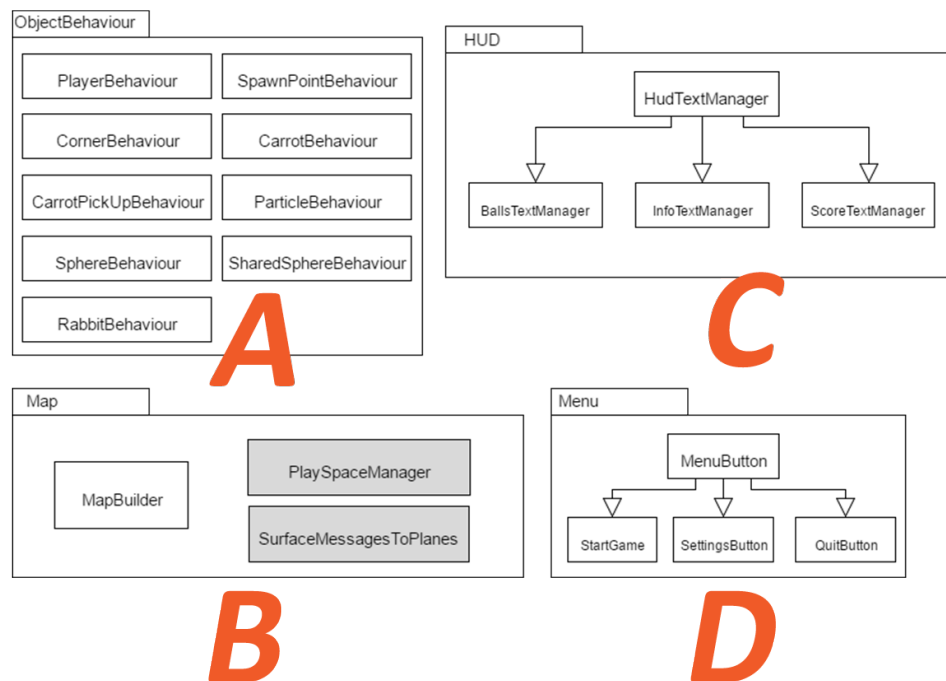


Figure 4.3: Folder A contains all the behaviour scripts that control certain in-game objects. Folder B contains the MapBuilder class which creates the map. The grayed out classes are from Microsoft and provide the MapBuilder information about the current room. This information determines the placement of the map. Folder C shows the inheritance of the HUD classes. Folder D shows the inheritance of the menu buttons.

HoloToolkit GitHub this functionality broke. Issue #70 talks about this problem (<https://github.com/Microsoft/HoloToolkit/issues/70>). For now nobody is capable to resolve this issue and therefore we chose to create an extra option in the menu to override the IP address stated in the original game build. This offers the user the option to say the IP address and join a custom IP address. It is not an optimal solution, but after discussing this with our client we all agreed that this is sufficient.

4.4. Photon Unity Networking framework

We not only used the Microsoft Sharing Service to share data between the HoloLenses, we also used the Photon Unity Networking (PUN) framework. This is a framework that enables developers to create real-time multiplayer games for Unity. We used this service mainly to share messages about the game (see Figure 4.4). Whenever the player scores, he/she will send a message about the current score to the opponent. The same happens with other data like amount of power-ups, amount of balls etc. This way both HoloLenses know what the other player has and can make decisions based on them. One example would be the amount of power-ups that the users are carrying. When one of the two users already carries one power-up, then no new power-up will be spawned anymore.

The Photon server is configured in such a way that the first player who joins is the so called 'master client'. This player basically creates the map, creates the game objects and controls everything in the game. The other player only sends messages when the score changes or when a bunny is hit. The master client will receive all this data and use it to calculate new statistics, make decisions and communicates this back to the other player.

Next to sending and receiving data, will the Photon server also synchronize all the ball objects. Those objects move quickly around and need to be synchronized real-time. The Microsoft Sharing Service was not able to do this accurately enough. That's is why this is the only object that is being shared using the Photon network. This way the amount of computations of the two networks are also more balanced and divided equally.

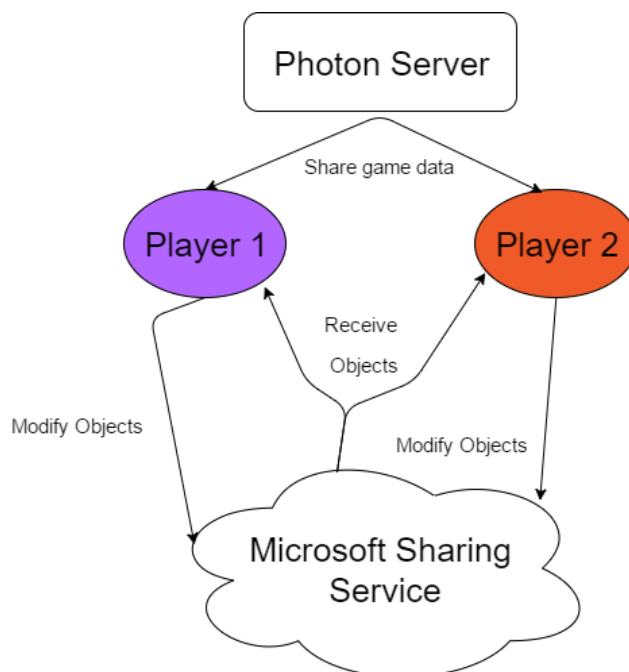


Figure 4.4: The Microsoft Sharing Service manages the objects that are currently in the game. A player can individually modify objects on the server. After a certain modification both players receive an update from the server. Exchanging other game data is handled by the Photon server. Photon makes it possible to directly send data to the other players in the room.

5

Code quality assurance

5.1. Software Improvement Group

On the 2nd of June we uploaded our provisional code to the Software Improvement Group. This group contains software evaluation experts who check your code on different characteristics. They look at how maintainable your code is, the quality of it and if it is readable for example. We got the following reply:

“De code van het systeem scoort 4,5 ster op ons onderhoudbaarheidsmodel, wat betekent dat de code bovengemiddeld onderhoudbaar is. De hoogste score is niet behaald door een lagere score voor Unit Size.

...

De aanwezigheid van test-code is in ieder geval veelbelovend, hopelijk zal het volume van de test-code ook groeien op het moment dat er nieuwe functionaliteit toegevoegd wordt. Over het algemeen scoort de code dus ruim bovengemiddeld, hopelijk lukt het om dit niveau te behouden tijdens de rest van de ontwikkelfase.” (see Appendix [D](#))

The second, also the final, upload will happen on the 26th of June. That is unfortunately also the deadline for this thesis, so we are not able to put the feedback of the second evaluation here. We did however improve our code based on the first feedback. Long methods are split into smaller ones and we made sure that every method has its own responsibility. Also more unit tests are added and the whole system is documented.

5.2. Automated Static Analysis Tools

We used multiple automated static analysis tools (ASATs) for this project to maintain good code quality. Visual studio comes with build in tools such as FxCop which detects possible violations concerning design, localization, performance, and security. Besides this we also added an extension called ReSharper. This tool contains many code format rules. This way the coding style of everybody in the team is the same and consistent. This does not only improve code quality, but also the efficiency of our workflow. For example a general rule for C# is to place all public variables above private variables. When all members obey this rule it is much easier to find and understand the needed variables within a class. Not only style is evaluated by ReSharper. This tool gives many warnings for possible improvements to code quality. An example are warnings about unused variables and unreachable code. We have chosen to use so many tools, because we never programmed in C# and Unity before we started this project. ASATs can be critical in learning a language the right way. Logically this is very important when working on a team project.

5.3. Testing our system

In order to ensure that new changes would not break functionality, unit tests were implemented. We mainly tested our game objects through unit tests. Other code, such as behaviour and spawners, were much harder to test through unit testing. So for those scripts, we would run the game every time after introducing a change. This way we ensured that new code would not break anything (see Figure 5.1).



Figure 5.1: The list of all our unit tests. The last three test classes are folded out to show how what they exactly test.

5.4. Version control

Throughout the project we used GitHub as our version control platform. This made it possible to work at the same time on the game. We also made use of the pull requests option of GitHub. For each new feature we implemented we made a new branch. Whenever the code was finished, we would make a pull request. Another member first has to review and test the code, before it was merged into the master branch. This process went well. A stable version was ensured by using this method of new code integration. Also whenever multiple people worked on the same files, we had the option to see the differences and choose what we wanted to keep and what not.

Final product

In this chapter the final product is described. The final product is a Windows Store application for the HoloLens platform. The whole game process is described, with elaboration where applicable.

6.1. Timeline of the game

The game starts with a main menu. Here you can start the game, set a custom IP for the Microsoft Sharing Service or quit the game. After you start the game, the current room will be scanned. After scanning the room, a playing field is placed. Both players have to stand in their own half to start the first round. A round ends when the score limit is reached or when the time is over. A game consists of multiple rounds and the player can choose whether to play a next round or quit the game after a round ends. Figure 6.1 gives a good overview of all the game states.

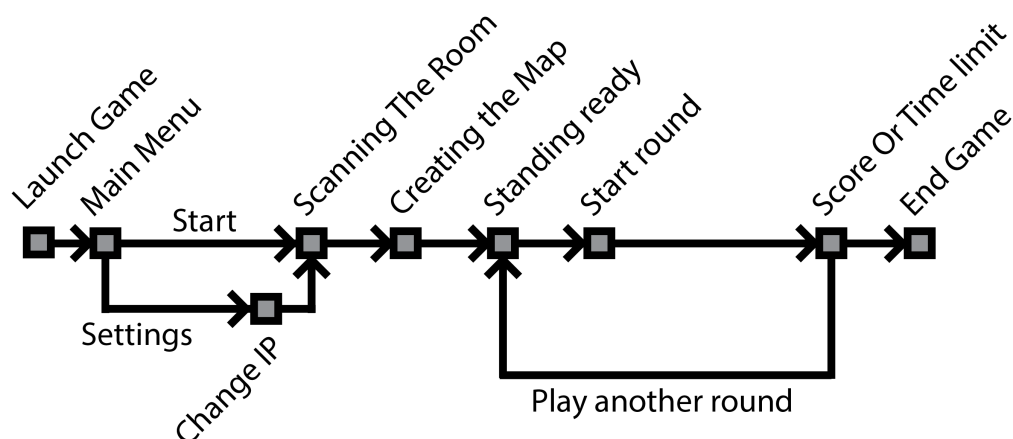


Figure 6.1: The general timeline of the game.

6.2. Game field

When the user starts the game, the game will give instructions to first look around for about thirty seconds. This will give the HoloLens the chance to scan the room. Our game will use this data of the room, it will try to find a suitable open space on the floor to place the playing field on (see Figure 6.2).

This playing field has green edges. In the middle there is a line that separates the map in two equal halves. Whenever a player crosses this line, he will not be able to shoot balls or pick

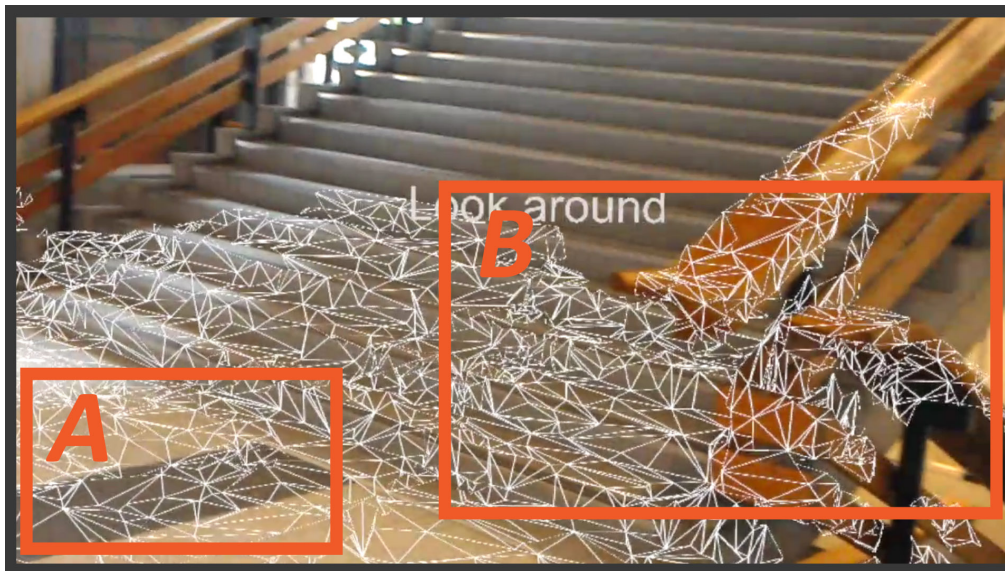


Figure 6.2: This is the wireframe of the room the player is currently scanning. Section A shows a part of the empty floor. In section B the stairs are scanned.

up any item. A text in the player's screen will pop up, telling the player that he/she should go back to his/her own side.

After the placement of the map, one circle will appear on each side of the field. These are so called spawn points. Both users get the instruction to stand in their own spawn point. When both players are standing in their circle, the game will start. The game has rabbits, balls and carrots. We will elaborate each of them below.

6.3. Players and their controls

The person playing the game is literally the player in the game (see Figure 6.3). The position and orientation is used as input for the game. A virtual ball is added in front of the player to show from which position the ball is shot. The opponent's player is also just the other person playing the game. Besides moving and rotating, the player can use the tap-gesture to shoot a ball. Speech recognition is used for other extra abilities, because the HoloLens has only two predefined gestures. Only the tap-gesture can be used in applications, the other, the bloom-gesture, is for exiting the current application. Defining custom gestures is hard, definitely in the circumstances of our game where people move all the time. The tap-gesture is therefore much more stable to use.

6.4. Shooting balls and their behaviour

People are able to shoot balls by making the air tap gesture. This is a gesture where the player holds his fingers in front of him/her. If you go with your wise finger to your thumb, then such an air tap gesture is made (see Figure 6.4). If the HoloLens registers this move, the game will fire a ball for you. When the ball hits a rabbit, the player will get points. All balls that lie on the ground can be picked up by standing on them.

The first player has everything in purple colors, while the second player has orange colors. The balls will also leave a trail of sparkles behind them. This makes the tracking of the ball easier for the player and makes the game more visually pleasing.

The HoloLens already has the data of the room. This means that all moving virtual objects will collide with real world objects. If there is a table in the middle of the room and you shoot a ball on top of it, then it will bounce off the table for example. This will create a more realistic blend of the real world with the virtual objects.



Figure 6.3: Both players standing on their half of the field. The players self can see the augmented reality, but an outsider can not see this.

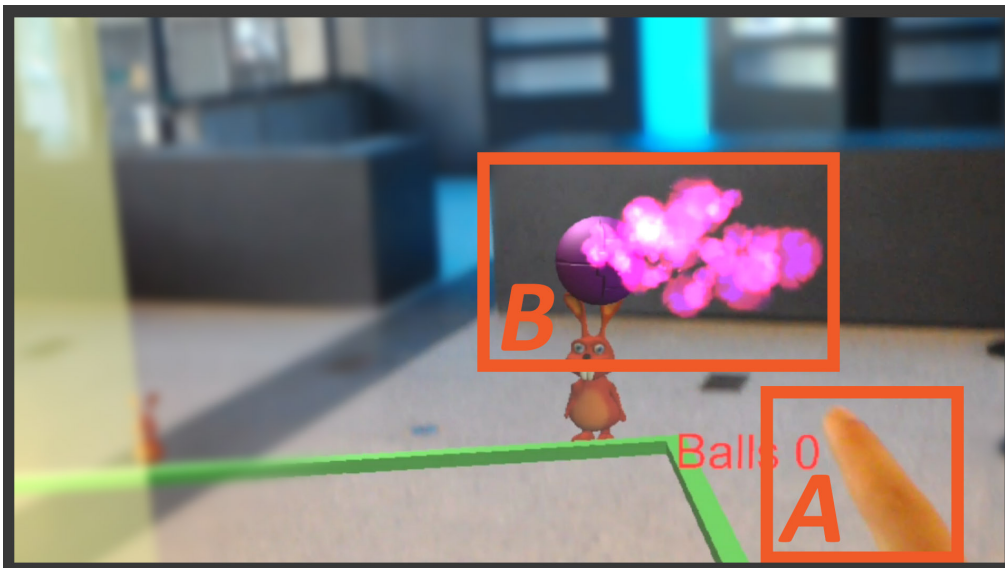


Figure 6.4: This is a frame just after player one shoots a ball. This is how the user sees the game. Section B shows the ball with the trail of sparkles. Section A shows the finger that made the tap gesture.

6.5. Targets

When the game starts, there will be three rabbits on each half. The rabbits are standing next to each other on the back of the map (see Figure 6.2). Both players have their own rabbits to defend. When a rabbit is hit, it will shrink and disappear. You will get points for hitting the opponent's rabbit. After some time they will appear again so players can continue to score. You are also able to defend your rabbits by standing in front of them and catching the ball that is going towards them.

6.6. How to win the game

The objective of the game is to be the first to score a certain amount of points. The person that is the first one to reach a certain score, will be the winner. Every round also has a time limit. If the time is over, then the player with the highest score is the winner. If the players

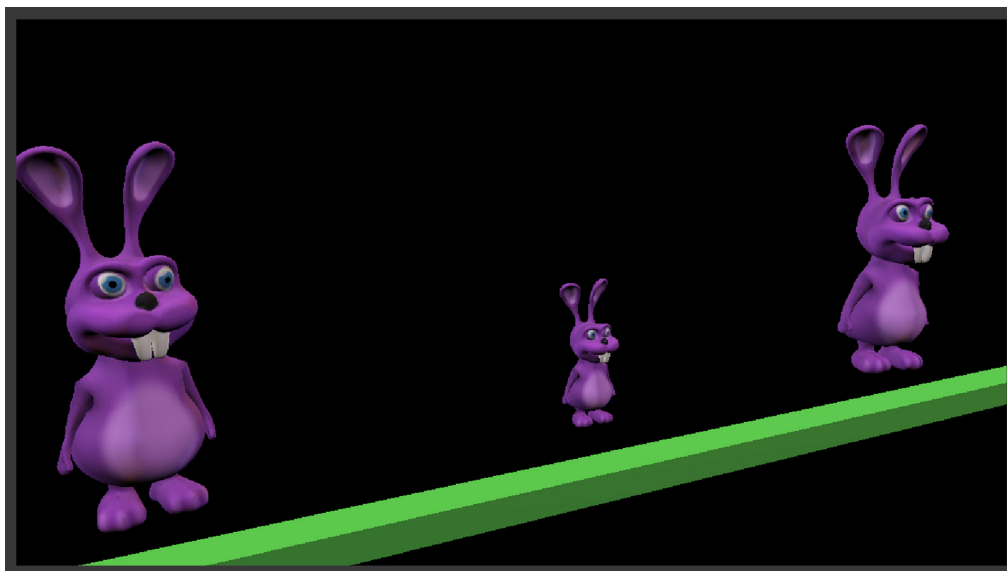


Figure 6.5: The three targets from team purple.

have the same amount of points, then this will result in a draw.

The player is able to score points by hitting the rabbits of the opponent with balls. There are rabbits in different sizes, the smaller a rabbit is, the more points you will get for hitting it.

6.7. Picking up power-ups and activating them

We made the decision to implement power-ups in order to make the game more dynamic. Every now and then a carrot will be spawned at a random place in the map (see Figure 6.6). If the player picks this carrot up, he/she is able to activate it by shouting “Carrot!”. This will spawn a carrot in the color of the player who activated it on the opponent’s half. One of the rabbits will randomly walk towards this carrot. This gives the player who activated the power-up an advantage, because the rabbit is much easier to hit now. The opponent has the option to pick up this carrot and then to reuse it. The rabbit will walk back to his original place in that case.

As can be seen from the screenshot below, there is a light beam shining on the carrot. We added this feature to make it easier for players to detect a power-up. The viewing angle of the HoloLens is really limited, so by adding a spawn sound and the light beam we tried to make the user more aware of the power-up.

6.8. Different playing strategies

The player is able to use different strategies. One can for example play really offensive. Constantly standing close to the middle line and shooting multiple balls at the same time is an example of such a strategy. This makes it hard to defend and the defender will need to move fast in order to catch all the incoming balls. This leaves no time for the defending player to score. The limited amount of balls that the player has, makes this offensive play not easy. Another strategy would be the defensive strategy. In this strategy the player would be the one standing close to his rabbits, trying to defend them. The person can catch balls and wait till he has a lot of them before trying to hit the opponent’s rabbits.

Players in the game are also able to pick up balls. This can be rewarding when those balls are used to score, but when a player is collecting balls they have to leave their targets undefended. This forms a trade-off. There is also the option to pick up power-ups. If you possess and activate the carrot power-up, a carrot will spawn at a random location and the opponent’s rabbit will walk towards it. This gives you the opportunity to score more easily. To defend this power-up the opponent can choose to run towards the spawned carrot and

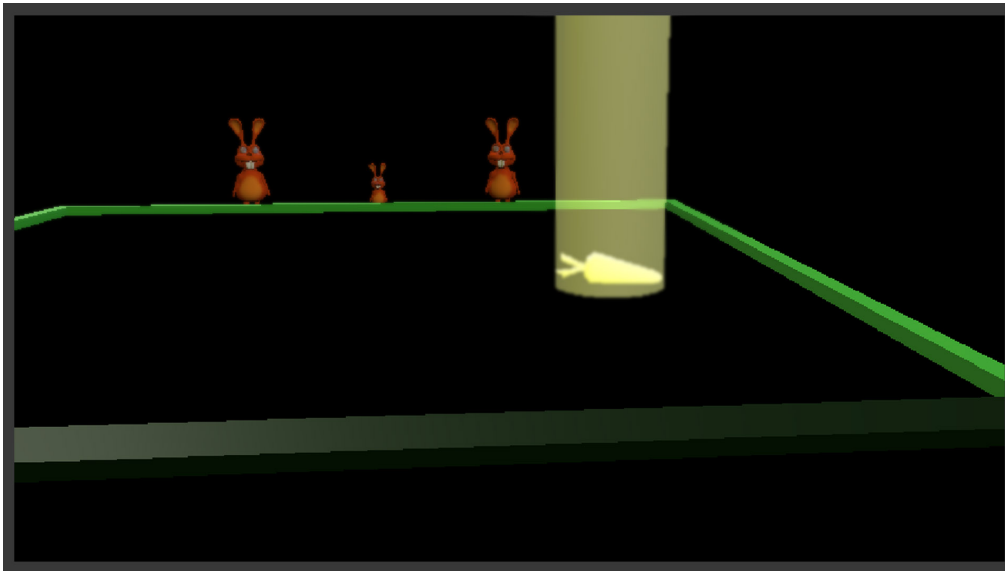


Figure 6.6: The yellow light beam with in the middle a carrot, this is the power-up. Randomly spawned at a location on the playing field.

pick it up before his rabbit does. This power-up will also create more randomness and allow players to use different strategies to win the game.

6.9. The value of sounds

Microsoft describes that the usage of different sounds in the HoloLens can be really valuable. Especially for the HoloLens, where they have the option to use spatial sound. This means that the sounds can come from certain directions with different volumes.

By adding spatial sound to ball bounces we make the players focus more on the balls and their behaviour. For this game it is important to follow the balls to score and prevent the opponent from scoring. By playing a sound when the ball bounces on the real world we also emphasize the room where you are standing in and how you can use this to your advantage. It is also more realistic that whenever the ball hits something, it makes a sound. This will get people quicker engaged in the game.

Whenever a power-up spawns, we also play a spawn sound from the direction of where it spawned. This way the user knows where a power-up has spawned relative to himself, even when the power-up is out of sight. As described, the HoloLens has a very limited viewing angle, therefore many objects are hidden for the user. This makes it hard to keep track of all objects around you. By adding sounds to objects on certain interactions, the user is better capable of understanding everything that happens. This is partly solving the problem of the limited field of view.

Game evaluation

This chapter will give an evaluation on the game. There were two main evaluation periods where different people tested our game and provided feedback. We will evaluate the demo at the International Festival of Technology (IFoT) and our own user study.

7.1. International Festival of Technology

The yearly International Festival of Technology was held this year from the 7th till the 9th of June. The goal of this festival is to inspire technology, music and art enthusiasts in Delft, the region and worldwide by demonstrating what Delft has to offer and to make technology accessible to everyone. Our client invited us to attend the Research Exhibition and gave us the opportunity to show an early version of our game as a demonstration.

People were generally interested in the project and engaged with lots of enthusiasm in the game. People of all ages walked by and used our game. There was an group of German high school students for example, but also an elderly man who was interested in augmented reality. We saw that everyone, no matter which age, got engaged in the game and played it with a lot of fun. This might be biased though, because the festival mainly attracts people with a technical background.

7.2. User study and results

On the 22nd of June we did a user study with fifteen volunteer students. They all played the game for at least two rounds and then filled in our version of the Game Experience Questionnaireimp (GEQ) [9]. This questionnaire is from the paper 'Measuring the Experience of Digital Game Enjoyment' where the focus is on characterization of game experience (see Appendix B). With a large range of measurements this questionnaire can reduce uncertainty associated with measuring a single modality. This results in increased validity, robustness and wider applicability of the total set of measures [8].

The main goal was measuring how engaged people are in the game. Simply put: "Is the game fun to play?" Another sub goal is to find out how much physical activity the game enforces.

7.2.1. Approach

Before playing the game we gave the participants a short instruction (see Appendix F). Then they played two rounds. One round takes up to 5 min if the score limit is not reached earlier in the round. The first round was mostly about learning and understanding the game and the controls. The questionnaire was administered immediately after the game had finished. We mainly used the core and post-game modules of the GEQ for our assessment. These probe the feelings and thoughts of the players while playing the game and afterwards. The results can be found in Appendix C. The most important results are visible in Table 7.1 and Table 7.2

Scoring scale	
0	Not at all
1	Slightly
2	Moderately
3	Fairly
4	Extremely

Table 7.1 The scale used for evaluation.

Core Module	
Competence	1,72
Sensory and Imaginative Immersion	1,90
Flow	2,44
Tension/Annoyance	0,89
Challenge	1,73
Negative affect	0,88
Positive affect	3,13
Post-game Module	
Positive Experience	1,86
Negative Experience	0,52
Tiredness	1,87
Returning to Reality	0,93

Table 7.2 The final results of the user studies.

7.2.2. Core module

We scored high on ‘Positive affect’ and ‘Flow’ and low on ‘Negative affect’ and ‘Tension/Annoyance’. These are good results and inline with our intended goals. Although ‘Competence’ scores a bit low. This is justifiable by the fact that some components of the game are not easy to learn in a few minutes. We found out that the build in tap gesture of the HoloLens is hard to understand. Directly starting the game without practising this gesture is not ideal. This gesture requires some prior practise. The second problem was the limited view angle that the screen of the HoloLens provides. This limitation makes it hard to keep a good overview of everything that happens in the game. The third problem is the delay in communication between the lenses. Sometimes an enemy ball is visible after a delay of roughly a second. This makes it hard to follow the game and defend your targets properly. Those reasons are probably why the ‘Competence’ component scores a bit low.

7.2.3. Post-game module

The tiredness score of 1.87 is not outstandingly high. The reason for this is obvious, we only played 2 rounds of a few minutes. The first round was to warm up and get to know the mechanics of the game. When watching the players, we observed that almost every player moved a lot from the moment that they understood the game. Afterwards a part of the players were even somewhat exhausted, because they were really competitive. When this game is played twice as long or multiple rounds are played, it becomes much more exhaustive. After only two short rounds, a score of 1.87 for tiredness is definitely not bad.

Overall the user study was useful for feedback on how to improve the game experience. We got a clear insight in which components of the game were hard to understand. Also the last open question of the user study confirmed a lot of our expected downsides. Many of them originate from limitations of the HoloLens. For example players said that:

- The lens is too heavy for a sport containing much movement.
- The limited viewing angle was frustrating.
- The screen becomes blurry when moving too fast.

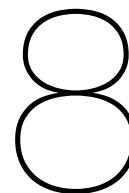
7.2.4. Specific questions and their relevance

In this section we will comment on specific questions from the GEQ. Some are perhaps not really applicable for our game or in the contrary questions that are very important.

Question 37 'It felt like a victory' This question is probably more meant for a story mode game. One where the user can always win. In the case of a competitive game 1 vs 1 there is most of the time a clear loser and winner. This automatically leads to very divergent answers.

Question 3 'I thought it was fun to play' Naturally this is a very important question, because one of the main requirements was to make the game fun to play.

Question 45 'I felt weary' This question is also important to meet the main requirement of making a game involving physical activities. We already explained why in general the tiredness score was not as high as expected. Even with this aspect taken into consideration, the score of this question was remarkably low: 0.7 on average. Afterwards we found out that many of the Dutch students who participated did not know the right definition of 'weary'. This likely causes an erroneous result.



Conclusion

The last few months we have been working on VRabl, an augmented reality sports game, that offers an engaging experience and motivates people to be physically active. We did research for the first two weeks in order to define a fun sports game, create ideas and look at existing literature and games. Based on the research we created a concept and implemented the sports game for the Microsoft HoloLens. With only two people we had to put a lot of effort in the process in order to create a fully working AR game, but in the end it was definitely worth it.

The game satisfies all of the 'Must Haves' and some of the 'Should/Could Haves' of the previously defined requirements in Appendix E. The weekly meetings with our client helped us to improve and make the game more fun. This could also be seen at our demo at both the IFoT and user studies. People got immersed into the game, sometimes forgetting that they were actually standing in a public place with an audience. Many were competitive and some even sweating after playing two rounds. This shows that the game encourages players to be physically active. Everyone who played VRabl had lots of fun. The game really enforced the players to move. Only after they stopped playing the game, they realized that they were tired and spent a lot of energy in movements. This shows that VRabl stimulates people to be more physically active. Players are rewarded with more points and a greater chance of winning when putting more physical effort into it.

VRabl is a front runner of multiplayer AR games and AR Superhuman Sports. In both fields there are still many challenges to overcome and improvements to be made. Our game shows what is currently possible and what the benefits are of such a game. To answer the research question, we believe that VRabl is an example of a game that will stimulate people to be more physically active while having fun at the same time. Although VRabl is not ready for an official release, it gives a good picture of what is possible. With more resources and time, it will definitely be possible to create a popular and stable augmented reality sports game. We believe that in a few years time, when the AR industry is more mature, games like VRabl will become more popular. New sports will arise, involving technology for human augmentation and enhancing human abilities.

9

Future work

We could say that the aim in the future is to create games with physical activity or Superhuman Sports, that become popular enough to participate in their own competitions. To make this possible some steps need to be taken. In this chapter we will explore what is going to happen and what needs to happen in order to achieve this goal.

9.1. Required improvements

There are many improvements in different fields that can be made to further improve augmented reality technology and the development for AR technology.

First of all there are certain limitations of the HoloLens and the development tools that are currently available. Microsoft is planning to let other companies create hardware for their AR operating system in order to let the AR market grow. They already have third-party hardware partnerships with Acer, Asus, Dell, HP and Lenovo [12], since many users complained about the small view of the HoloLens during the user studies. Other aspects are the blurry colors that appear when moving your head too fast, the weight of the glasses and the battery duration. Those aspects should be improved in order to make augmented reality sports games more attractive.

There are also many improvements that could be made for VRabl. The game has only one power-up, the carrot power-up. The game would be more dynamic and fun when there are more power-ups. Here you have to keep in mind that the game should not be too difficult to explain or too chaotic to play. So maybe one or two more would be a nice addition. A should have that we did not have time to implement were different kinds of balls. This could be a future improvement for the game. Also more targets and perhaps defensive objects could be added to make the game more dynamic and create more strategies. The map size could be made dynamic to the environment. There are many visual aspects that can be improved. Unfortunately, we did not have much time left to make the game visually pleasing. Due the limited time we used a free unity model of a rabbit and only designed a few custom 3D objects that were necessary for the game. The current version of VRabl shows a concept of an AR Superhuman Sport and the possibilities within such a game.

9.2. From Superhuman Sports DreamTeam to the Tokyo Olympics

The field of Superhuman Sports still needs time to develop. The Superhuman Sports Society, which was established in 2015, plans in making more people aware of this new concept. They will also be talking to organizers of the 2020 Tokyo Summer Olympic games and hope to showcase some new sports there. This will hopefully let the concept of Superhuman Sports grow.

Towards this goal, several Superhuman Sports design competitions are planned with the first competition in July 2018 in Delft. With this design competition in mind, it is the plan to setup a DreamTeam in Delft. The team for that game can obtain information and learn from the experiences that we gained during this project.

9.3. CHI PLAY student game design competition

We will also participate in the CHI PLAY 2017 student game design competition. This edition will be held in Amsterdam, which is really convenient for us to actually attend the conference. This competition will provide us the opportunity to showcase our game and designs. We will write a paper, create a video of the game and give an actual demo of the game during the conference. The game will be reviewed on different criteria, including originality, quality and the positioning and articulation of the game's contribution. The submission deadline for this competition is on the 30th of June, which means after the deadline of this thesis. So unfortunately we will not be able to show the results here.

10

Discussion

This chapter will discuss several aspects of this project. The first one are ethical aspects related to our sports game. Second are some challenges that we had to overcome regarding our team. Last are some technical aspects.

10.1. Ethical aspects that need to be considered

There are some ethical aspects that should be discussed when using the game. The game uses two Microsoft HoloLenses in order to project the virtual game on the real world. The HoloLens has multiple input devices and sensors, such as the camera, GPS and microphone. This automatically brings privacy and security issues [17]. The HoloLenses first need to scan the entire room before the game can start. When you scan a room, the HoloLens will save this room as a 3D-model of many meshes. So if you are for example using the game in your living room, then a model of your living room will be made and stored on the HoloLens. Our game uses this data to create the playing field and place it in a open space. We are not using the data for any other purpose.

The mapping of your room officially also belongs to Microsoft. They store this data on the HoloLens and maybe even on their cloud. They use it to keep track of screens and holograms that you have placed in the room. Next time when you start the HoloLens, the same holograms and screens appear at the same place as where they have been placed. If the HoloLens gets hacked or stolen, then the hackers will have this data. They will gain access to the location of your room and the model of your room. You definitely want to avoid that. The location and mapping of the room is sensitive data and should be handled carefully.

The HoloLens is also capable of filming. If you are in a room with other people and you film what you are seeing through the lens, then those people will be recorded. Here the same ethical issues arise as with the video camera. Some people don't want to be recorded and then you are breaking their privacy. Microsoft choose to implement a small white light which indicates when the device is recording. This makes it harder to film someone unnoticed. This is very import when these kind of glasses become more and more popular. In that stage people will wear them in public places. Perhaps it even becomes unlikely to walk through the streets without being filmed.

All these ethical issues not only hold for the Microsoft HoloLens, but for all AR glasses. It is important to consider AR privacy in this early stage, while the technologies are still young and malleable [17].

10.2. Project team

Normally the Bachelor Project will be executed by a group existing of three to four members. Initially we registered for this project with four students. One of the four already warned us in advance that he might not be participating due to personal issues. It turned out this was the case, so we started with three members instead of four.

The first two weeks we had to come up with ideas and setup everything required for the project. When setting up a repository, Trello and creating an initial project in Unity we found out that one of the members had trouble to keep up with everything. We found out that he did not have the required knowledge for this project. He was a Spanish exchange student and had a totally different background, he had taken more electrical engineering than computer science courses. We spent a considerable amount of time explaining everything to him so he could keep up with the progress. After some conversations with the bachelor project committee, it was decided that this project was not suitable for his background. Halfway week 3 we were left with only 2 of the initial 4 students. This meant that we had to leave some initial ideas behind, because we did not have the time to implement all of the proposed features. At this point in the project we rescheduled everything and made a planning that was more feasible for us. In Appendix G we describe the process in detail. Eventually with some extra effort we managed to implement all the 'must haves' in the game.

10.3. Automated integration testing

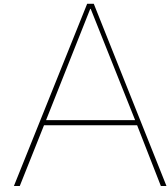
Normally you would write lots of different tests in order to ensure a correctly working system. For this project we mainly wrote unit tests. It was really hard to find something suitable to test Unity objects and their behaviours. We did find a testing framework for higher level testing called 'Unity Test Tools'. The only problem was that the last update was somewhere in September 2016. We tried to use it, but many parts were not functioning properly in our version of Unity. So we decided not to use it and test the more visual components manually. Using version control and different branches we managed to constantly maintain a thoroughly tested and working version on the master branch.

10.4. Augmented technology

The Microsoft HoloLens is still in development. There is no commercial version on the market yet. They released the developers edition a few years ago. Developers can get used to the hardware and learn how to program it. Unfortunately there is not much good documentation available for now. There are a few Microsoft tutorials that teach you certain aspects of the HoloLens by creating a game. Many of these tutorials just give you example code of how they implemented it. It was very hard to implement something different from their version. The code quality of the example code of Microsoft that was used in the tutorials is really bad. Methods of more than 500 lines and magic numbers made it hard to understand. The lack of information and documentation forced us to put extra effort and time into the project in order to get things working.

10.5. Release of the game

The game itself is currently stable. Running the game however, requires some technical knowledge and is not so obvious. A local Microsoft Sharing Service needs to be setup for every game. This server needs a computer to run on. After running the server, the player is able to connect the game using the settings button in the main menu. When the game starts, it automatically connects to a Photon cloud server. So we need to make sure that this cloud server is always running in order to play the game. So unfortunately the game is not release ready for the Windows store. It should be made more user-friendly and automatically connect to the needed servers before an official release is possible.



Research report

A.1. Introduction

The first two weeks of this project are dedicated on researching the topic and the possibilities. In the introduction (of the thesis) we gave a detailed and extended description of the topic and scope of this project. In order to fully explore the possibilities of creating a final product that meets all requirements, we need some insights in the current technologies and existing frameworks.

In section A.2 the problem will be described, analyzed and defined. We will also give more context and information concerning the problem. Possible solutions to the problem will be given in section A.3. In the last section, section A.4, we will give an outline of our approach and the tools that we will be using.

A.2. Problem definition & analysis

Console gaming is getting more and more popular. Partly to the increasing convergence of new technologies and their possibilities [3]. There are clear downsides to this shift of popularity. One study showed that children who play moderate amounts of electronic games weigh more than their counterparts that played less [21]. There is also a significant positive relationship between physical activities and cognitive functioning [18]. Physical activities are a great part of their general development, especially for young children and teenagers.

So how can we create games, involving physical activities, more attractive in a world where there exists a plethora of console games that are fun to play without any physical effort? To answer this question, we will try to come up with a game with physical movement. One where the physical efforts add extra fun and motivates people to play the game.

A.2.1. Specific requirements

The client demanded some minimal requirements for the game. The game should be a Superhuman Sports game using mixed reality. Superhuman Sports are activities that rely on technology for human augmentation to enhance a human ability, involve physical fitness and skill, and are played for fun, competition or health reasons [10]. Mixed reality implies a world where physical and virtual objects exists at the same time. It is a mix of reality and virtual reality (VR). Next to these characteristics the game also need to meet the following requirements:

- It should be fun to play and engage/motivate people to do sports
- One player versus another player
- Competition based (Points, time, performance etc.)
- Time limit of 30 minutes per round, including explanation time

- The needed space should not be bigger than a regular sports hall
- It should be easy to explain and understand (Max. 10 minutes)
- It should enhance the player's physical abilities

To meet these requirements and answer the research question, we need to come up with a fun game using augmented reality (AR) and involving physical activity. There are multiple challenges that need to be addressed in order to create such a game. We will briefly amplify the most important challenges.

A.2.2. Fun factor

One of the challenges is to create a game which is fun to play. A game that is enjoyable for people of all ages. In this part we will research which aspects contribute to making a game fun. Perhaps there are general rules that can make a game fun. In that case we could consider these factors while designing and creating the game. Maybe there are other aspects, depending on the game itself, that make a game enjoyable.

Chris Crawford looks at this problem by going back to the history of games. The games currently available are so complex and diverse that it is hard to indicate a single clear function. Finding the origin of human games is hard, but perhaps looking at other animals will suffice. Many young mammals play games. They wrestle with each other or silently approach a target. These games are an important way of obtaining hunting skills. Chris Crawford claims that the fundamental motivation for all game-playing is to learn something [4].

We can say that learning is an important aspect of games. There are also many other aspects that need to be considered in order to make a game more enjoyable. We can divide most of these aspects in three main categories: challenge, fantasy, and curiosity. In the next parts we will explain how a game can achieve this.

Challenge

According to Thomas W. Malone, challenges in games engage a person's self-esteem, and thus making it more captivating [11]. Offering challenges and the opportunity to learn something is sufficient to create a motivating game that is fun to play [5]. Challenges offer the player the ability to learn new skills. Therefore challenge is important, because learning is a truly fundamental motivation for playing games [4]. To make sure the game contains enough challenges, we will need to consider the following aspects that make a game challenging [11]:

A simple and clear goal

Such a goal is important to make sure the player directly understands the game. This offers the player the ability to reason about the rules as well.

Multiple difficulties

Different difficulties offer every player the opportunity to play at his/her level. Players who are good don't get bored and bad players will not find it too challenging. Games are most enjoyable when they contain the right amount of challenge. One where the player is able to win by putting in a certain amount of effort.

Useful goals

Running from point A to B or solving math questions is generally not considered as something fun to do. Although bringing gold from point A to B or being able to fly a rocket to the moon by solving math questions is much more fun. In the second example the assignments are much more useful and rewarding. Performing a job using a particular skill, without any specific goal, tends to get boring.

Multiple goals

Building a level which consists of multiple goals gives the player more challenges. In our case of a 1 vs 1 game it is important that both players are having fun. Every round or game there

is inevitably someone going to lose. By creating multiple goals both players can succeed in small parts of the game. This ensures that both players stay engaged and like the challenges created by the game.

Hidden information

Hiding information from the players will cover the optimal strategy. This is an important aspect, which not only creates an extra challenge, but also creates room for fantasy. The next section will describe fantasy more in-depth.

Randomness

Adding a random aspect to the game makes it unpredictable. This directly creates a challenge for the player. Random changes in a level can create difficult situations. In those moments a dynamic strategy and good anticipation is rewarded. When a game does not contain any randomness it could become too straightforward and boring.

Fantasy

“A very important motivation to play games is fantasy fulfillment” [4]. Games can include fantasy in a physical way or as a social situation. Fantasy demands the player to empathize in a given situation. This situation does not have to be real or credible. A physical example is a game where the player is extremely strong. A social example is a game where the player plays a role and owns an entire empire. Games with emotional fantasies are likely to be more popular [11].

The fantasy needs to be appealing to the potential player. We do not focus on a particular specified group. Therefore we could choose a general fantasy that is very appealing to all, but this is a hard task. A better option would be to create multiple fantasies and let the player choose their favorite. Another possibility is to allow the player to project his or her fantasy into the game. A game could support this by allowing the player to make custom changes. For example the naming or coloring of a character in game.

We can distinguish two types of fantasy: extrinsic and intrinsic fantasy. Extrinsic fantasy depends on whether the skill is used correctly. With extrinsic fantasy the fantasy only depends on the skill, but with intrinsic fantasy the skill also depends on the fantasy. When shooting at targets in a game, the amount of targets still standing in the world depends on the player's shooting skills. This would be an example of intrinsic fantasy. Thomas W. Malone says that it is almost always better to use intrinsic fantasy compared to extrinsic when trying to make games more fun and compelling [11].

Curiosity

The scene where the game takes place is important. The environment of a game should not be too simple. Otherwise there is no room for the player to imagine what could come next. The environment should also not be too complex and overwhelming so that the player does not understand it. When the environment has the right complexity it raises expectations by the player. Curiosity can be induced by changes in the environment. It is also possible to do this on a higher level with structures and patterns in the given information. Audio and visual effects are a clear example of the first group. These effects can be used as decoration or enhance fantasy to become more involved in the game. On the other hand we have the high level of curiosity. These kinds of curiosity can be induced in different ways [11]:

- **Completeness:** You give, for example, the player all puzzle pieces except the last few. The last pieces are needed to figure out what the total picture is.
- **Consistency:** When a game follows a certain pattern for a given time and then suddenly changes this pattern. These inconsistencies will raise the player's curiosity.
- **Parsimony:** For example when there is a general rule in the game. Every now and then the player gets a hint based on this general rule. The player has to figure out by themselves what the underlying rule is.

A.2.3. Augmented reality

The game needs to use some sort of technology that supports mixed reality. The aim is to encourage people to participate in a game that requires physical movement. To achieve this goal we first have to make a choice of what kind of technology we want to use. The most widely used technologies are augmented and virtual reality. Augmented reality adds virtual content to the real world where virtual reality replaces the real world with a virtual one [20].

Augmented reality is more suitable for games involving movements. Simply because augmented reality does not block your view of the physical world. This makes movement such as running much more natural and safe. It will also feel much more natural when your environment changes according to your movement. A virtual reality glass will block your sight, which also prevents social interaction with a team member or the opponent.

There are many different technologies for augmented reality. The Microsoft HoloLens is one of them. Our client has two of them available for us to test and develop on. This is the main reason why we chose to work with the HoloLens. The most important features of the HoloLens that we have identified so far are: virtual objects creation, gestures interaction and mapping environments. People are also able to view the same objects at the same time [22]. This makes it possible to add new components to an existing sport. We will aim for a sport disguised as a game. The need of movement is less obvious and will not discourage people in participating, but in order to win you still need to move around.

Microsoft HoloLens

The HoloLens has many features and possibilities. Some are already embedded and supported, while other features can be easily extracted from the available sensors. Our main focus is on the predefined features which work best. A good working and stable solution is better than a complex and not stable solution. For that reason we will try to keep the basics as simple as possible and we can also add additional modules if needed. The standard features of the HoloLens are [13]:

- Detection of the entire real-world room where the user is standing in. This spatial mapping is basically a virtual 3d representation of the real-world.
- Placing custom virtual objects in the real world. Objects can be attached to 3d coordinates in the real world, as can be seen from Figure A.1. It is also possible attach objects to the screen of the HoloLens as some kind of overlay.
- Gaze is the ability of the HoloLens to recognize the direction that the user is looking in. This feature can also be used as input of the user, for example for selecting objects.
- The HoloLens has many predefined gestures. This allows the user to interact with the environment.
 - Tap: Comparable to a mouse click. Tap in combination with gaze offers the ability to click at any place in the world.
 - Hold: This can be used to pick up objects.
 - Manipulation: This feature offers the manipulation of holograms. A object can be resized or rotated by the use of manipulation.
 - Navigation: This is comparable with a virtual joystick. The feature allows for velocity-based continuous scrolling or zooming.
- Voice input allows the user to give voice commands to the system.
- The HoloLens can manipulate sound in such a way that the user thinks that it is coming from a certain direction from the environment. This is called spatial sound. Spatial sound gives the ability to make objects perceivable for the user even if they are out of sight.
- Recognize points in the real world that are saved in the HoloLens. This so called spatial anchors make it possible to recognize locations in the real-world across from previous sessions.

- The current location and orientation of the users is also known.

The communication between two HoloLenses is also possible. The lenses can distribute data such as recognized points with their location. Also all the game data is shared so if player A moves an object player B is able to perceive this.



Figure A.1: Example of RoboRaid played on the Microsoft HoloLens [1]

Existing augmented reality games

There are already existing augmented reality games. Some of them are available for the HoloLens. In this section we will name three examples. We will analyze them and see what we can learn from them.

Holo Pool Hall

This HoloLens game simulates a pool hall in which the player shoots the ball by a vocal command: "SHOOT". When we want to create a game with balls or bullets, we could learn from this game. The balls have nice physics and bounce off the wall when they hit it.

We also could implement our own vocal commands for specific actions. A player can switch between different guns/weapons or activate special abilities by saying specific words for example.

RoboRaid

A very fancy and well-developed shooting game created by Microsoft and widely used as a demo for the HoloLens. There are three features worth mentioning:

- Mapping: At the start of the game, the HoloLens maps your surrounding. It knows where to fix holes on the wall, that will spread insect-robots. If two players are playing a sport in which there is interaction, we need to map them continuously. It will be necessary to implement a real-time mapping functionality.
- Interaction with virtual objects: This is a good example in case we want to create a game that involves dodging certain objects. In this game you need to dodge fireballs for example.
- Fun: Like we explained before, fun is an important aspect of a good game. This game has interesting objects that surround the player. This will engage the player even more in the game.

Pokemon GO

This is one of the first big "augmented reality" games that really became successful globally. It is taking advantage of the popularity of Pokemon. The goal was for players to go outside and catch certain "Pokemon" monsters. The application used your camera to display Pokemons onto the real world. Those could be spotted in different locations and were dependent on the environment. Near water you would spot certain water Pokemons for example. This is a good example of an AR game, even though the application is not using holograms.

A.2.4. Exertion Games Lab

The Exertion Games Lab is researching different kinds of future games. Exertion games involve physical effort [16]. They also contain physical, mental and social benefits [15]. The lab focuses on interactive experiences and in particular on games where the human body is the centre of the experience. Their projects are games that use some sort of physical activity and the latest technologies, often including augmented reality. Since this is relatable to what we will be working on, we will pick some their projects to analyze and learn from them.

Hanging off a Bar

In this game the player hangs onto a physical bar. Underneath the player is a digital river, which motivates the player to hang on. Once in awhile a raft floats beneath the player, so he/she has some recovery time. This recovery time and the frequency of the raft appearing will decrease over time [14]. While the activity of hanging is really simple, players still got captivated by the game. The developers asked the players for feedback and got useful comments. Better graphics, facilitating fearful emotions, hanging together and a virtual audience were a few ideas that emerged [14]. These are aspects that we can also put in our game. A game is more fun when there are other players present. Also facilitating strong emotions makes a game more interesting.

Joggobot

The Joggobot is a computer-controllable flying Quadrotor. When you go jogging and can't find a partner, then you can take the Joggobot with you. The robot is a companion that is able to distract the runners from their exhaustion and challenge them to increase their effort [6]. Here they also asked users to give feedback and come up with improvements. People prefer the bot to react to their speed and actions. Also people got annoyed when the bot was flying too fast for them and some did not like the design [6]. From this project we can learn multiple things. People like to get reactions based on their own actions, this will make any activity more interactive. You should also take the design into account, a design can make a game appealing or unappealing to a given target group.

A.3. Possibilities

When trying to create a proper idea we need to consider multiple aspects. First, the game needs to meet the client's requirements. Secondly, it should be feasible to create it within the time span of this project. Thirdly, the game should be original. It should have new aspects added by augmented reality and if possible it could even be a whole new concept. Last but not least the final product should be a game which encourages people to perform physical activity and is fun to play for everybody.

A.3.1. Ideas

It is an option to create a new game from the ground up. We decided to build on an existing sport or game to make sure we already have many aspects that people consider as fun. One of the requirements for the game is that it should be easy to understand for everybody. Therefore the second reason is that an entire new sport is hard to explain and takes much time to learn.

We came up with many different ideas. A short description of the most promising ideas will be given below. The first few are inspired by existing sports and the second group mostly by existing games. For each idea the playfield, players and the rules will be explained. Then we will shortly explain how we think that augmented reality can enhance the ideas.

Tennis

The playing field is a flat field where a virtual net splits the field in two equal parts. There will also be a virtual ball. Both players will be holding a virtual racket and stand on their own side of the field. The scoring system would be similar to a normal tennis game.

Volleybal

The field is similar to tennis only the net is higher. There will also be a virtual ball. Both players have again their own half, but here the players will use their hands to hit the ball over the net. The rules and scoring system will be the same as with real world volleyball.

Dodgeball

The basic field would be a flat field with only one line separating the field in half. Both players have their own side and throw virtual balls at each other. The rules are similar to that of dodgeball. Here again we have the option to add extra features to the field and balls.

Fencing

There are no clear restrictions to this playing field. Both players will be carrying a virtual sword. Score can be gained by hitting the enemy with this sword.

Shooter

With this game we have the option to either divide the field in two halves or choose to have one big field. Both players will carry a virtual weapon that can shoot virtual balls, bullets or other objects. Score can be increased by hitting the other player using your weapon.

Target Shooting

The map consists of a standing position for each player and a space for targets. Both players have a ranged weapon. The players aim for target every round. Hitting specific targets will increase the score of the player.

Base Defense

The map is made up of two parts that are exact copies of each other. Both containing a start and end point connected by a road. The players have the ability to build objects to defeat the virtual enemies that travel from start to end. The players also have some kind of virtual money. The score is decreased when a virtual enemy reaches the end of the road. A player can prevent this from happening by building objects that defend their road. Every round there comes a so called wave of enemies, which the player need to stop. In addition to defending the player can also spend money and effort on offense to make the next round harder for the opponent.

Angry Birds

The map consists of two bases and two shooting areas. Both players can shoot different objects to the other base. Score is gained by destroying the other player's base. Collision of the projectile with the enemy base can cause it to (partly) collapse.

Pong

The field requires two goals where the players can score and a ball. The two players have their own virtual object which bounces the ball. To score a point the ball has make it to the other goal. Blocking the ball will cause it to bounce into the other direction.

A.3.2. Enhancement with augmented reality

For every game mentioned above augmented reality can add extra aspects. In general every game could benefit from a dynamic environment. The looks of the floor can be altered to make the game more challenging or make it look better. In some of the above games a ball bounces on the surface where the bouncing properties can be changed according to the virtually created appearance. Many ideas are influenced by the strength of the player. We could temporary enhance this or add enhancements that will last for the entire game.

Temporary enhancement are a form of power ups or abilities. This can add to the fantasy aspects that Thomas W. talked about [11]. Using abilities and powerups in a smart way can be awarding to the player. This decision process stimulates the player's fantasy.

For many ideas the HoloLens could add extra virtual players in the game that will help one of the players or make it harder for both. For example a bot that bounces all balls nearby to the other side. This bot can help in a tennis, volleyball or pong game. For dodgeball a similar bot can shoot balls at the players and try to score. The balls and projectiles can also be altered. Since they are virtual, we will have the freedom to change them in every possible way. The speed, size and movement can be changed. Also certain balls/projectiles can bring more points if they hit the right targets.

A.3.3. Downsides

Popular sports such as tennis and volleyball are not really unique or special to play. When the added features of the HoloLens are not impressive enough the game is probably not motivating enough for people to play. A sport like dodgeball or fencing is less known. Only fencing has the problem that virtual objects cannot give feedback of the sword. A move cannot be blocked, therefore the only option is to make the player fight against virtual enemies instead of each other. The clear downside of this change is that the competitive aspect is less notable.

Games like Target Shooting, Angry Birds and Base Defense all lack the physical activity. By adjusting the game we could force more movement in the game, but it is likely that this is not natural. It could demotivating when people notice that they are forced to move every time in order to achieve some goals. We think it is better if the physical activity is naturally embedded in the game for example with a shooter. A game like Pong, in contrast, is all about movement. The bouncing of the ball requires the player to stand at the right place all the time. This directly forces physical movement. This physical movement is way more obvious than in a shooter. For people with some kind of aversion to physical activity, a game of Pong is not considered as fun.

Some of the games are less well-known for most of the players. A game such as Base Defense needs a lot of explaining. One of the requirements is that the game is easy to understand, so these games are probably not a good starting point.

A.3.4. Comparison

To get a better overview of all the ideas we will compare each idea based on the most important requirements. The first aspect is physical activity. Here we have to see whether the original game contains enough movement aspects. The second demand is making a game which is easy to understand. The third requirement is making the game fun to play. This requirement is hard to measure, therefore we will also look at how unique the game is. When a game has nothing new to offer, it will not really motivate people to play it. Eventually for the last aspect we will look at how suitable the game is for the HoloLens. As already mentioned above there are certain limitations to the HoloLens. Figure A.2 below shows each requirement and the suitability for each game, while Figure A.3 explains how we established these ratings for the more complex aspects.

There is not a single idea that fits all conditions. When combining two ideas we are able to achieve higher scores on multiple aspects. We think a combination of a shooter with dodgeball will be able to fit all the requirements. It's basically a dodgeball game but instead of throwing balls you will be shooting objects. Creating some sort of gun and detecting the shot projectiles is possible with the HoloLens. This removes the downside of dodgeball. When players have their own half, we will also not have the problem of close range chaos of a shooter anymore. The virtual shooting aspect will add an unique feature to a real life physical game. This way we have an unique and suitable physical game for the HoloLens. The game is also easy to explain and understandable because most people already know dodgeball and have seen or played a shooter. Figure A.4 below shows the amount of unique global unit sales of different Call of Duty games as of January 2017. This is only an example of how big one shooter is, there are many more other shooters available.

Different aspects				
Idea	Physical activity	Easy to understand	Fun/Unique	Suitable for the HoloLens
Tennis	+	+	-	+/-
Volley-ball	+	+	-	+/-
Dodgeball	+	+	+/-	+/-
Fencing	+	+	+	-
Shooter	+	+	+	+/-
Target Shooting	-	+	-	+
Base Defense	-	-	+	+
Angry Birds	-	+	+	+
Pong	+	+	-	+

Figure A.2: Rating of each aspect for each game

Different aspects		
Idea	Fun/Unique	Suitable for the HoloLens
Tennis	Almost everybody knows this sport	It is hard to recognize the intended
Volley-ball	Almost everybody knows this sport	It is hard to recognize the intended direction of the pass
Dodgeball	This sport is mostly played at schools. Not many know there exists a real competition	Recognizing the throwing movement is hard
Fencing	Not a well known sport, most only know it from the Olympic Games	No feeling of feedback of hitting an obstacle or the opponent
Shooter	Shooters are well known console games, but in real life much more rare	When both players are at close range the detection of the gun and projectiles is hard. The HoloLens works best at a distance of a few meters
Target Shooting	It is possible to do this at every fair	There are no real design problems with this game. There are already HoloLens demo games containing target shooting
Base Defense	A real life Base Defense game is very unique	There are no real design problems with this game
Angry Birds	A real life Angry Birds game is very unique	There are no real design problems with this game
Pong	Pong in real life is unique, but comparable with sports like table tennis	There are no real design problems with this game

Figure A.3: Aspects analyzed more in-depth

A.4. Plan of action

For this project we are going to use an agile planning. We will be using weekly sprints. At the start of every week we will decide which tasks need to be done for that specific sprint. On the first day of each sprint we will evaluate and test all tasks that are done in the previous sprint. For these weekly processes we will be using Trello. At the start of the project we will create an entire backlog of all the tasks that need to be done in order to create the final product.

A.4.1. Organization

It is always good practice to organize a project. In this project we will be using different tools and platforms to stay organized.

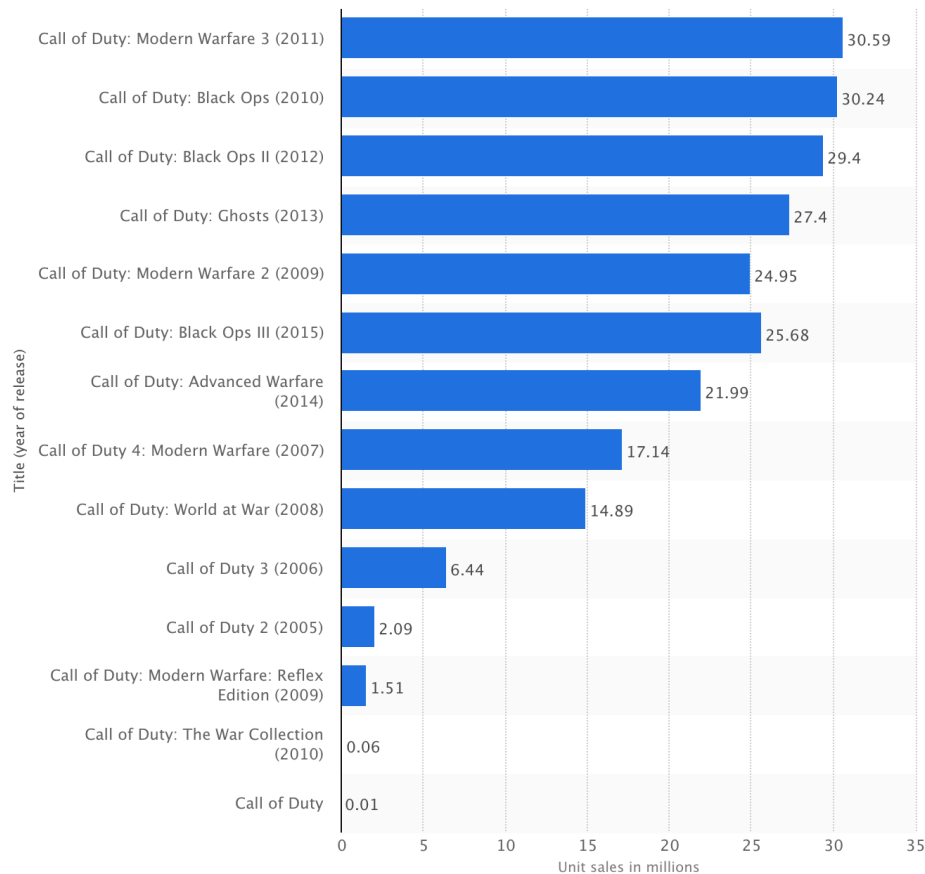


Figure A.4: Statistics of the amount of Call of Duty games sold [19]

For our code we will be using git. It gives you the option of version control, which means you can keep track of changes in your code. It will also make it easier to program simultaneously and merge the code when needed. We have chosen to create a repository on Github. It gives developers the option to create pull requests and do code reviewing before a feature is implemented, which will increase the code quality. We have decided to use a private repository, keeping an eye on the Superhuman Design Competition at TU Delft at the end of 2019 [10].

There will be many different tasks which need to be divided over the group members. We decided to use Trello to keep track of these tasks. It is basically a task management app with a nice visual overview. We will be tracking which tasks need to be done, which are currently being tackled and which are done. It also gives you the option to categorize the tasks and assign people to each task.

Further we will be using Google Drive to save all the general documents. We will be using Overleaf to work on all \LaTeX documents. Whatsapp and daily meetings will be used for mutual communication between group members. Email and weekly meetings are used for communication with the client, TU coach and the bachelor project committee members.

A.4.2. Code quality assurance

Obviously high code quality is important for a good game. When we find out during development that our requirements need to change to fit the initial goal it is important that we can do this anytime. Therefore the code needs to be structured, contain enough comments and have high cohesion and low coupling.

To make sure we have the needed code quality we will use static analysis tools. Those tools will provide warnings during development that improve the quality. We will use Fxcop and

CAT.NET to detect Code violations. For the format and style we will use StyleCop. Eventually for overall detection of smells and duplicate code we will use SonarCube. Besides those tools we will also do two submissions of our code to the Software Improvement Group (SIG). They will evaluate the code quality and provide feedback on how we can improve it.

A.4.3. Testing

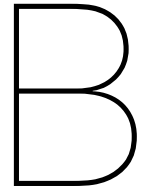
For our code we will not only use static analysis tools to ensure high code quality, we will also be using different levels of software testing.

First we will be using unit tests to test small individual parts of code. It will increase the confidence in changing and maintaining the code. If a change is introduced that will break anything, the unit tests will automatically detect this.

After writing unit tests and having multiple connected units, we will need to write integration tests. These tests are able to expose defects in the interaction between multiple integrated units. With these tests you can ensure that the different units within the game are correctly integrated.

When we know that all units are correctly integrated, we are able to perform system tests. In our case, it will mean running the whole game and see what goes wrong/is not working. This ensure that your game is fully functioning as intended.

Last but not least, we will be using acceptance tests to see whether the product meet all requirements and specifications. Some of the requirements can be tested directly by the developers. Others only by asking real world users to participate in a test. The final product should be fun for people of all ages. This means that we will also need to do different user studies. We can measure different aspects with these tests, such as the intuitiveness or the amount of fun of the game. For these tests we can ask friends, family and even strangers to participate and give feedback.



User study questionnaire

B.1. Game Experience Questionnaire – Core module

Please indicate how you felt while playing the game for each of the items, on the following scale:

- 0. not at all
- 1. slightly
- 2. moderately
- 3. fairly
- 4. extremely

The Questions:

- 1. I felt content
- 2. I felt skillful
- 3. I was interested in the game's story
- 4. I thought it was fun
- 5. I was fully occupied with the game
- 6. I felt happy
- 7. It gave me a bad mood
- 8. I thought about other things
- 9. I found it tiresome
- 10. I felt competent
- 11. I thought it was hard
- 12. It was aesthetically pleasing
- 13. I forgot everything around me
- 14. I felt good
- 15. I was good at it

16. I felt bored
17. I felt successful
18. I felt imaginative
19. I felt that I could explore things
20. I enjoyed it
21. I was fast at reaching the game's targets
22. I felt annoyed
23. I felt pressured
24. I felt irritable
25. I lost track of time
26. I felt challenged
27. I found it impressive
28. I was deeply concentrated in the game
29. I felt frustrated
30. It felt like a rich experience
31. I lost connection with the outside world
32. I felt time pressure
33. I had to put a lot of effort into it

B.2. Game Experience Questionnaire – Post-game module

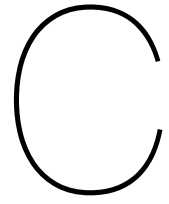
Please indicate how you felt after you finished playing the game for each of the following items:

0. not at all
1. slightly
2. moderately
3. fairly
4. extremely

The Questions:

1. I felt revived
2. I felt bad
3. I found it hard to get back to reality
4. I felt guilty
5. It felt like a victory
6. I found it a waste of time
7. I felt energised

- 8. I felt satisfied
- 9. I felt disoriented
- 10. I felt exhausted
- 11. I felt that I could have done more useful things
- 12. I felt powerful
- 13. I felt weary
- 14. I felt regret
- 15. I felt ashamed
- 16. I felt proud
- 17. I had a sense that I had returned from a journey



User study results

C.1. Table with ratings

Question:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Person 1	4	4	4	4	4	1	1	4	4	3	2	1	3	3	0	3	1	0	4	2	1	3	1	1	4
Person 2	4	2	3	4	3	0	0	1	4	2	2	4	4	3	0	3	2	0	4	2	1	3	0	4	4
Person 3	2	3	3	3	2	0	0	0	3	2	3	2	3	1	0	3	2	3	3	3	1	3	1	1	3
Person 4	3	2	4	4	4	1	0	1	3	3	2	4	4	2	0	2	4	3	4	2	1	0	0	4	3
Person 5	3	3	4	3	4	0	0	0	2	2	3	3	3	1	0	2	2	3	4	1	0	0	0	3	3
Person 6	3	2	3	4	3	0	0	1	1	2	3	3	3	1	1	2	4	4	4	1	1	2	0	4	3
Person 7	3	1	4	4	4	1	1	1	0	2	2	1	3	4	2	2	2	2	4	2	2	2	3	2	3
Person 8	3	2	4	3	3	0	0	0	3	3	2	2	3	1	0	1	3	3	4	3	0	0	0	2	2
Person 9	3	3	4	4	4	0	0	1	2	2	2	4	4	2	1	2	3	2	3	0	2	3	2	3	
Person 10	2	3	3	3	3	2	1	2	3	3	1	2	3	3	2	3	1	1	3	2	2	2	2	3	2
Person 11	2	2	4	3	3	0	1	0	3	3	1	1	3	2	0	1	1	1	3	2	1	1	0	0	2
Person 12	4	3	4	3	3	0	0	0	0	0	2	2	3	1	0	2	2	0	3	3	1	0	0	0	1
Person 13	2	3	2	3	1	0	0	0	2	3	1	0	1	1	0	0	0	0	2	1	0	0	0	0	1
Person 14	2	0	2	3	2	1	1	0	3	3	2	2	1	0	1	1	1	1	2	0	2	1	1	3	3
Person 15	2	3	4	3	3	0	0	0	3	0	2	2	3	3	0	2	3	2	4	3	0	0	0	0	2
Question:	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	
Person 1	3	3	1	3	1	0	4	2	0	0	0	3	0	3	3	3	4	0	3	1	0	1	1	0	
Person 2	2	4	1	3	3	0	4	2	0	0	0	2	0	3	2	1	3	1	2	3	0	0	1	0	
Person 3	2	3	1	2	2	0	2	2	0	0	1	2	1	1	3	1	3	2	3	2	1	1	2	2	
Person 4	4	4	2	4	1	1	2	3	0	0	0	2	0	3	3	2	3	1	3	2	0	0	3	3	
Person 5	3	2	0	3	2	0	1	1	0	0	0	1	0	3	2	0	0	0	1	1	0	0	2	1	
Person 6	4	3	1	4	3	1	2	3	0	0	0	3	0	3	3	0	0	1	4	0	0	0	1	3	
Person 7	3	3	3	4	2	2	4	1	4	1	1	4	2	3	3	2	1	2	4	1	1	1	4	4	
Person 8	3	2	0	2	2	0	1	2	0	0	0	0	0	1	2	0	0	2	0	0	0	0	2	2	
Person 9	3	3	1	3	3	3	2	2	1	2	2	2	2	3	3	2	1	1	3	1	0	1	3	2	
Person 10	3	3	2	2	3	1	3	2	2	2	1	2	1	2	3	2	2	2	2	0	1	3	2	3	
Person 11	2	1	0	2	0	0	2	1	0	0	0	2	0	0	2	0	0	1	0	0	0	0	2	0	
Person 12	4	3	0	4	3	0	0	2	0	0	0	1	0	3	4	0	0	0	1	0	0	0	0	1	
Person 13	2	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
Person 14	3	3	3	2	2	1	2	0	1	0	0	1	1	1	0	1	0	2	1	0	0	0	1	2	
Person 15	3	2	0	2	1	0	0	0	0	0	0	2	0	2		0	0	1	0	0	0	0	0	0	

Number	Question
1	I felt content
2	I felt skilful
3	I thought it was fun to play
4	I was fully occupied with the game
5	I felt happy
6	It gave me a bad mood
7	I thought about other things
8	I found it tiresome
9	I felt competent
10	I thought it was hard
11	It was visually pleasing
12	I forgot everything around me
13	I felt good
14	I was good at it
15	I felt bored
16	I felt successful
17	I felt imaginative
18	I felt that I could explore things
19	I enjoyed it
20	I was fast at reaching the game's targets
21	I felt annoyed
22	I felt pressured
23	I felt irritable
24	I lost track of time
25	I felt challenged
26	I found it impressive
27	I was deeply concentrated in the game
28	I felt frustrated
29	It felt like a rich experience
30	I lost connection with the outside world
31	I felt time pressure
32	I had to put a lot of effort into it
33	I felt revived
34	I felt bad
35	I found it hard to get back to reality
36	I felt guilty
37	It felt like a victory
38	I found it a waste of time
39	I felt energised
40	I felt satisfied
41	I felt disoriented
42	I felt exhausted
43	I felt that I could have done more useful things
44	I felt powerful
45	I felt weary
46	I felt regret
47	I felt ashamed
48	I felt proud
49	I had a sense that I had returned from a journey
50	Do you have any other comments, improvements or feedback? Feel free to say whatever comes to mind.

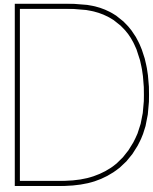
1. -
2. -
3. -
4. -
5. Misschien kunnen jullie nog iets duidelijker aangeven van de tussenstand is tijdens het spelen, zodat er wat meer competitie is, maar verder een nice game!
6. Misschien kunnen er nog wat sicke visuals bij als surroundings om de experience compleet te maken.
7. -
8. -
9. -
10. De bril was best zwaar en drukte super hard op mijn neus.
11. Het feit dat je maar een klein deel van je veld kon zien maakte het spel een stuk moeilijker en ik kan me voorstellen dat dit voor mensen een frustrerend aspect is.
12. -
13. -
14. Adjust the size of arena maybe a good feature
15. Couldn't understand the light beam that was used for the carrot, but it has to do with the FOV I think.

Core Module

Competence	1,72
Sensory and Imaginative Immersion	1,90
Flow	2,44
Tension/Annoyance	0,89
Challenge	1,73
Negative affect	0,88
Positive affect	3,13

Post-game Module

Positive Experience	1,86
Negative Experience	0,52
Tiredness	1,87
Returning to Reality	0,93



Software Improvement Group evaluation

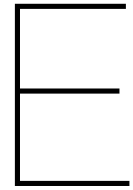
De code van het systeem scoort 4,5 ster op ons onderhoudbaarheidsmodel, wat betekent dat de code bovengemiddeld onderhoudbaar is. De hoogste score is niet behaald door een lagere score voor Unit Size.

Voor Unit Size wordt er gekeken naar het percentage code dat bovengemiddeld lang is. Het opsplitsen van dit soort methodes in kleinere stukken zorgt ervoor dat elk onderdeel makkelijker te begrijpen, te testen en daardoor eenvoudiger te onderhouden wordt.

In jullie project is MapBuilder.CreateCorners hier een voorbeeld van. Je kunt deze methode nog verbeteren door het bepalen van de coördinaten te scheiden van het aanmaken van de objecten. Let op dat jullie al erg hoog zitten qua score, dus dit zijn voornamelijk kleine puntjes en geen wereldschokkende aanbevelingen.

De aanwezigheid van test-code is in ieder geval veelbelovend, hopelijk zal het volume van de test-code ook groeien op het moment dat er nieuwe functionaliteit toegevoegd wordt.

Over het algemeen scoort de code dus ruim bovengemiddeld, hopelijk lukt het om dit niveau te behouden tijdens de rest van de ontwikkelfase.



Requirements using the MoSCoW approach

Aspects:	Must	Should	Could	Would not
Playable for 2 persons or more.	X			
Competition based.	X			
Every round takes 30 minutes or less, including explanation of the game rules.	X			
Playable in a space smaller than a sport hall.	X			
Easy to explain and understand (5-10 minutes).	X			
Includes some sort of physical activities.	X			
Players needs to shoot a ball	X			
Detect collision of the ball and the real world	X			
Player collision with ball	X			
Recognize location and orientation in the field	X			
At the end of a round there is a winner	X			
Place extra virtual objects in the world		X		
Player able to shoot different balls		X		
Game is easy to download and install if you have the required hardware		X		
Pausing or ending game earlier		X		
Different type of maps			X	
Sharing game results to social media			X	
Video tutorial of how to play the game			X	
Leaderboard of all highest previous scores.			X	
More than 2 players				X
Option to save the game and continue later.				X



User manual

F.1. Starting the game

Turn the Microsoft HoloLens on. When you see the app menu, find the game called 'VRabl' and use the tap gesture to run the game.

F.2. Game Menu

When the game starts you will see the 'Made With Unity' logo. After the logo disappeared, you will see the game menu. Here we have three options:

- **Start**
Tap this button to start the game
- **Settings**
Here you can say the IP of the Microsoft Sharing Service to connect to a specific server. An IP address exists of four numbers with periods separating them. Say each of the four numbers as an individual number. For example, when your IP is 192.168.45.49, you have to say: "one hundred ninety two", "one hundred sixty eight", "forty five" and "forty nine". The HoloLens will recognize them, automatically place periods between them and try to connect to the given server. When it failed to connect or you made a mistake, you have to restart the application and try again.
- **Quit**
Tap this button to quit the game

F.3. Standing ready

When the game is first started, you have to look around for approximately half a minute. This gives the HoloLens the chance to scan the room that you are in. Try to slowly move your sight around the whole room. After a while, the game field with two sides will appear. There will be one circle on each side, one orange circle and one purple circle. The game will instruct you to stand in one of those circles. When both players are standing in the right circle, the game will start.

F.4. In-game

When the game has started, the player is able to shoot balls. By holding your hand in front of your face and making the tap gesture, a ball will be shot. You have limited amount of balls. Luckily there will be white balls that randomly appear on the ground, those can be picked up by standing on them. Each player has three rabbits on his/her side. When you stand on the opponent's side, a warning will appear on your screen. You will also not be able to shoot balls or pick up any items. The goal is to hit the opponent's rabbit and to defend your own

rabbits. You can defend them by standing in front of them. If a ball is coming towards you, then you can block it by 'catching' it. This means that you just have to stand in front of the ball. This will also give you one extra ball.

There are so called carrot power-ups that will randomly appear on the field. If you walk on it, then you will pick up the carrot. When you say 'carrot', a carrot will spawn on the opponent's side and one random rabbit will walk towards it. This will make it easier for you to hit the rabbit. When the opponent spawns a rabbit on your side, you can also pick it up and avoid that your rabbit walks the whole way towards that carrot. Hitting the small rabbit with a ball will give 50 points, the bigger ones are worth 10 points. The first player that scores 80 points has won the round. If nobody reaches this score, then the player with the highest score after 5 minutes will win. If the score is equal, it will be a tie.

F.5. After game

When one round is over. It will ask the first player if she/he wants to play another round. If the player shouts 'yes', then a new round will start, the steps will be starting again from point F.3 (Standing ready). If the player says 'no', then the game will be exited



Process evaluation

This chapter will give an evaluation on the whole process of the project. The plan of action in the research report described what we wanted to use to stay organized. This chapter will evaluate our planning, external and internal communication.

G.1. Planning

As described in the plan of action in the research report, we mainly used Google Drive and Trello for planning. In Trello we have a backlog of what needs to be done in the upcoming two weeks. Every week there was a general overview of every task had to be done that week. Whenever somebody is working on a task, he has to indicate that on Trello so the others know that the task is taken care of. There is also the 'Bugs' list on Trello, where all current bugs are listed. When working on a particular feature, bugs that does not require immediate attention can come to light. It is very useful to have a bug list at those moments. Furthermore we used Google Drive in order to work collaboratively on documents. If we needed new ideas or discuss old ones, we would use Google Docs to do this. This certainly was efficient and effective. It enabled us to work remotely and helped to keep a clear communication.

G.2. External communication

There were multiple parties involved in this project. We had weekly meetings with our client, Stephan Lukosch. Those meetings were really useful. In the first few meetings we mainly discussed the goal and ideas for the game. After the research period we tried to show a prototype every meeting. Lukosch was able to see the progress and give feedback, suggest improvements and come up with new ideas. We also had contact with our coach, Elmar Eisemann. In those meetings we mainly discussed different ideas. Our coach had some really creative ideas that we used as inspiration for our game.

G.3. Internal communication

We tried to physically meet as much as possible during the week. We worked mostly at a central location on the campus (e.g. EEMCS faculty or the TU Delft Library). This made it easier to make sure that everyone is actively involved in the project. It also helped to solve emerging problems, and increased communication effectivity for important decisions.

When we were not able to meet physically, like in the weekends or whenever somebody was not available during the day, we used Telegram to communicate. We tried to update each other as much as possible. Bugs that required immediate attention were put on Telegram. This in contrast to the bigger bugs, which sometimes took days to fix, that were posted on Trello. Important bugs had a higher priority and were directly tackled by our team. We also put all other technical aspects, like errors, specific IDs and useful links of this project there. Using Telegram instead of Whatsapp has the advantage that you are not that much

distracted by other people. It also has a nice and clean desktop client, which was convenient to use while programming (Figure G.1).

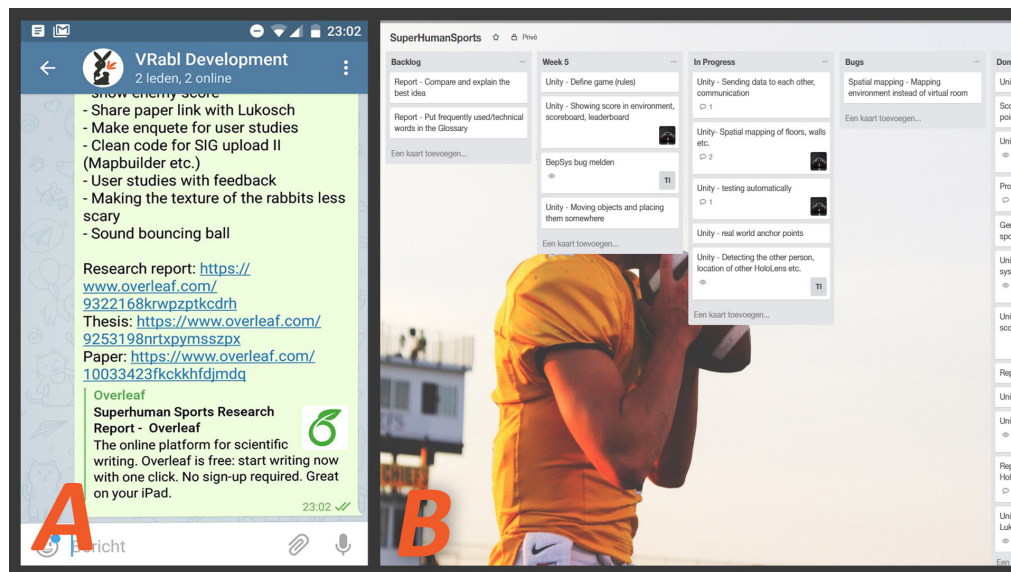


Figure G.1: Section A shows the project group on Telegram where all serious topics regarding our project were discussed. Section B shows our Trello page in week 5 of the project.

Bibliography

- [1] These are the first apps and games for Microsoft's HoloLens. <http://www.hotspotjournal.com/these-are-the-first-apps-and-games-for-microsofts-hololens/>, 2016. [Online; accessed 02-May-2017].
- [2] Bradmore and Magus. 2016 league of legends world championship by the numbers | lol esports, 2016. URL http://www.lolesports.com/en_US/articles/2016-league-legends-world-championship-numbers.
- [3] Jo Bryce and Jason Rutter. The gendering of computer gaming: Experience and space. *LSA PUBLICATION*, 79:3–22, 2003.
- [4] Chris Crawford. The art of computer game design. 1984.
- [5] Melissa Federoff. *Heuristics and usability guidelines for the creation and evaluation of fun in video games*. PhD thesis, Citeseer, 2002.
- [6] Eberhard Graether and Florian Mueller. JoggoBot: A flying robot as jogging companion. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 1063–1066, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1016-1. doi: 10.1145/2212776.2212386. URL <http://doi.acm.org/10.1145/2212776.2212386>.
- [7] Katherine Hollist. Time to be grown-ups about video gaming: the rising esports industry and the need for regulation. *Ariz. L. Rev.*, 57:823, 2015.
- [8] Wijnand IJsselsteijn, Wouter Van Den Hoogen, Christoph Klimmt, Yvonne De Kort, Craig Lindley, Klaus Mathiak, Karolien Poels, Niklas Ravaja, Marko Turpeinen, and Peter Vorderer. Measuring the experience of digital game enjoyment. In *Proceedings of Measuring Behavior*, pages 88–89. Noldus Information Technology Wageningen, Netherlands, 2008.
- [9] Wijnand IJsselsteijn, Yvonne de Kort, and Karolien. Poels. The Game Experience Questionnaire: Development of a self-report measure to assess the psychological impact of digital games. Manuscript in Preparation. in preparation.
- [10] Kai Kunze, Kouta Minamizawa, Stephan Lukosch, Masahiko Inami, and Jun Rekimoto. Superhuman sports: Applying human augmentation to physical exercise. *IEEE Pervasive Computing*, 16(2):14–17, 2017.
- [11] Thomas Malone. What makes things fun to learn? heuristics for designing instructional computer games. In *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, pages 162–169. ACM, 1980.
- [12] Rob Marvin. Microsoft's mixed reality plans go far beyond hololens | pcmag.com, 2017. URL <https://www.pcmag.com/article/352880/microsofts-mixed-reality-plans-go-far-beyond-hololens>.
- [13] Microsoft. Development, learn the fundamentals of holographic development. <https://developer.microsoft.com/en-us/windows/mixed-reality/development>, 2017. [Online; accessed 01-May-2017].
- [14] Florian Mueller, Cagdas Toprak, Eberhard Graether, Wouter Walmink, Bert Bongers, and Elise van den Hoven. Hanging off a bar. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 1055–1058, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1016-1. doi: 10.1145/2212776.2212384. URL <http://doi.acm.org/10.1145/2212776.2212384>.

- [15] Florian 'Floyd' Mueller, Martin Gibbs, and Frank Vetere. Taxonomy of exertion games. In *Proceedings of the 20th Australasian Conference on Computer-Human Interaction: Designing for Habitus and Habitat*, OZCHI '08, pages 263–266, New York, NY, USA, 2008. ACM. ISBN 0-9803063-4-5. doi: 10.1145/1517744.1517772. URL <http://doi.acm.org/10.1145/1517744.1517772>.
- [16] Florian 'Floyd' Mueller, Darren Edge, Frank Vetere, Martin Gibbs, Stefan Agamanolis, Bert Bongers, and Jennifer G. Sheridan. Designing sports: A framework for exertion games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2651–2660, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979330. URL <http://doi.acm.org/10.1145/1978942.1979330>.
- [17] Franziska Roesner, Tadayoshi Kohno, and David Molnar. Security and privacy for augmented reality systems. *Commun. ACM*, 57(4):88–96, April 2014. ISSN 0001-0782. doi: 10.1145/2580723.2580730. URL <http://doi.acm.org/10.1145/2580723.2580730>.
- [18] Benjamin Sibley and Jennifer Etnier. The relationship between physical activity and cognition in children: A meta-analysis. *Pediatric Exercise Science*, 15(3):243–256, 2003. doi: 10.1123/pes.15.3.243. URL <http://dx.doi.org/10.1123/pes.15.3.243>.
- [19] Statista. All time unit sales of selected games in Call of Duty franchise worldwide as of January 2017. <https://www.statista.com/statistics/321374/global-all-time-unit-sales-call-of-duty-games/>, 2017. [Online; accessed 30-April-2017].
- [20] Rick Van Krevelen and Ronald Poelman. A Survey of Augmented Reality Technologies, Applications and Limitations. *The International Journal of Virtual Reality*, 9(2):1–20, June 2010.
- [21] Elizabeth Vandewater, Mi suk Shim, and Allison Caplovitz. Linking obesity and activity level with children's television and video game use. *Journal of Adolescence*, 27(1):71 – 85, 2004. ISSN 0140-1971. doi: <https://doi.org/10.1016/j.adolescence.2003.10.003>. URL <http://www.sciencedirect.com/science/article/pii/S0140197103000903>. Video Games and Public Health.
- [22] Dianna Yim, Garance Nicole Loison, Fatemeh Hendijani Fard, Edwin Chan, Alec McAllister, and Frank Maurer. Gesture-driven interactions on a virtual hologram in mixed reality. In *Proceedings of the 2016 ACM Companion on Interactive Surfaces and Spaces*, ISS Companion '16, pages 55–61, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4530-9. doi: 10.1145/3009939.3009948. URL <http://doi.acm.org/10.1145/3009939.3009948>.